Ingeniería Técnica en Informática de Gestión

PROYECTO FIN DE CARRERA

MÓDULO DE PERSONALIZACIÓN DE AVATARES EN JUEGOS MULTIUSUARIO ONLINE

Autor: Javier Echániz Bombín

Tutor: Telmo Zarraonandia Ayo

Leganés, Septiembre de 2015

Agradecimientos

A Telmo y Mario, por darme la oportunidad de hacer el proyecto, ayudarme y no olvidarse de mí en ningún momento.

A todos los compañeros de trabajo de la Fundación y de Delta, que me animaron y me ayudaron cada vez que acudí a ellos.

A mis compañeros de carrera y de universidad, por todas las horas compartidas entre clases, prácticas, biblioteca, entrenamientos, partidos y algún que otro viaje, porque nunca faltaron las risas por mal que se nos diese algo.

A mis amigos, por estar ahí siempre desde hace tanto tiempo pase lo que pase, porque éramos unos críos cuando nos conocimos y ahora con el paso del tiempo son nuestros críos los que se van conociendo.

A mi hermana y mi cuñado, por esa cisterna que me sirvió de pupitre mientras duraron las obras de su casa y en la que me pude preparar un par de asignaturas durante un verano. Por Noel y Jaime, esos dos terremotos que siempre son capaces de sacarme una sonrisa.

A mis padres, por todos los sacrificios realizados para poder darme la mejor educación posible, espero no haberles fallado. Por haberme dado la libertad de hacer siempre lo que me gustaba. Por esas palmeritas que compraba mi madre en época de exámenes porque había oído que el azúcar era bueno para el cerebro, no seré yo quien la lleve la contraria a estas alturas.

Y por supuesto, a Ana, por ser como es, dejarme ser como soy y aguantarme desde hace ya 14 años, porque no es tan fácil. Por servirme de motivación y picarme a su manera para terminar el proyecto y no dejarlo a medias. Y por María, por poder disfrutar de esa sonrisa permanente que desde hace un año hace que todos los días merezcan aún más la pena.

A todos ellos, gracias.

ÍNDICE

1. Introducción

En este proyecto de fin de carrera se ha realizado una aplicación que permite a los usuarios crear y almacenar un avatar de manera que dicho avatar pueda ser utilizado en posteriores ocasiones en diversos juegos multijugador online.

Aparte de la aplicación que permita crear y almacenar el avatar, también se ha desarrollado una interfaz que permita la integración del avatar en el motor de juegos OpenSpace[1] (dentro de los módulos de ejemplo de SmartFoxServer[2]), de manera que, a partir de un XML almacenado en la base de datos, el motor puede integrar dicho avatar en el juego.

Con esta personalización del avatar se persigue la identificación del usuario en el juego y con ello su inmersión [1] del usuario en el juego. Esto es especialmente importante en videojuegos educativos, donde la identificación del usuario con su avatar es una de las características de los juegos que favorece la aparición de un estado de flujo en el jugador [2], el cual a su vez favorece el aprendizaje y la asimilación de los conceptos tratados en el juego.

La aplicación permite al usuario modificar escoger el sexo del avatar, y modificar su rostro, cuerpo y vestimenta a su gusto en función de las distintas opciones que se hayan creado a tal efecto. Como se ha dicho anteriormente, la personalización del avatar persigue la identificación del usuario en el juego, y al igual que no todos los días se viste uno de la misma manera, es posible que no todos los días el usuario quiera tener la misma imagen de avatar, ya que dependiendo de cómo se encuentre, es posible que quiera mostrar una imagen distinta de un día a otro de sí mismo. Por ello, una vez ya se ha creado el avatar, existe la posibilidad de modificación del mismo (a excepción del sexo) por el propio usuario.

[1] http://www.openspace-engine.com/
[2] http://www.smartfoxserver.com/

El almacenamiento del avatar va a servir también para guardar los logros conseguidos por el usuario en los distintos juegos y sesiones que realice, de manera que se pueda seguir una evolución del usuario a lo largo del tiempo.

Actualmente podemos encontrar aplicaciones que permiten crear avatares personalizados, pero dichas aplicaciones en algunos casos solo permiten la caracterización de una de las partes del avatar sin contemplar el avatar entero, y en otros casos, permitiendo la caracterización del avatar completo, no permiten la opción de integrarlo en un motor de juegos de con efecto de visión 3d, ya que solo nos ofrecen una visión de 2 dimensiones.

1.1 Objetivos

Personalizar un avatar. Crear un módulo de personalización de avatares para su uso en juegos multiusuario online, en el cual, el usuario tendrá la posibilidad de modificar a su antojo aspectos como la forma de la cara, con distintos tipos de ojos, pelo, bocas, nariz y orejas, el color de piel o la ropa con que se vista.

Almacenar los resultados del juego. Crear una base de datos con las características del avatar de cada usuario, así como los diferentes resultados obtenidos en cada una de sus conexiones a un juego con el fin de poder evaluarlo posteriormente.

Integrar un avatar en los módulos de ejemplo de SmartFoxServer. Crear una interfaz que permita introducir el avatar creado en el módulo de personalización en los distintos módulos de juegos de SmartFoxServer para que a través del motor de OpenSpace el avatar interactúe con los distintos escenarios existentes.

Implicar a los usuarios en el juego. Con la visualización en tercera persona del avatar definido por el usuario se quiere implicar a dichos usuarios en el juego, de manera que actúen con su avatar como si de ellos mismos se tratase, y así conseguir que el fin educativo del juego sea más realizable.

Facilitar el aprendizaje del usuario. A través de la inmersión del usuario en el juego, se pretende que éste alcance un estado de flujo, con el cual el grado de aprendizaje sea cada vez mayor.

1.2 Estructura del documento.

En este punto se detalla la estructura y contenidos de los distintos capítulos del documento.

- **Descripción del problema**.

Se realiza una breve introducción acerca de los objetivos de este proyecto y cuáles son las alternativas existentes en este momento.

- **Metodología**.

En esta parte se detallan los diferentes tipos de metodologías existentes (predictivas, adaptativas), explicando en qué consiste cada una y se explica por qué se ha decidido seguir una metodología en concreto en la realización de este proyecto. También se detalla la planificación seguida en el proyecto, así como un presupuesto basado en la utilización de recursos necesaria para llevar a cabo el proyecto.

- **Estado del arte**.

Se explica la importancia de los avatares hoy en día y la necesidad de la sociedad en mantener esa información actualizada. Se describen algunas de las distintas aplicaciones que se pueden encontrar ahora mismo a la hora de crear avatares y los diferentes estilos de avatar que se pueden encontrar, realizando una comparativa entre distintas aplicaciones encontradas. Por último se realiza un estudio de tres aplicaciones con el objetivo de obtener datos para nuestra aplicación, y se describe el motor de juegos en el que deseamos integrar nuestra aplicación.

- **Análisis.**

En este capítulo se realizara una descripción detallada de la aplicación, así como el flujo de información y su tratamiento.

- **Diseño.**

Se describe el diseño realizado previo a la implementación de la aplicación.

- **Implementación y configuración**.

Se detalla la estructura de la aplicación y como debe ser la configuración para su correcta integración en el motor de OpenSpace. Se realiza también un estudio de las distintas herramientas tenidas en cuenta para la implementación del proyecto y la elección definitiva.

- **Herramientas y lenguajes utilizados**

Se describen las herramientas y los lenguajes utilizados para la realización del proyecto.

- **Gestión del proyecto**

Se describen las tareas en las que se ha dividido el desarrollo del proyecto y los costes relacionados con la realización del mismo.

- **Conclusiones y posibles mejoras**.

Contiene las diferentes conclusiones y opiniones surgidas al término del proyecto, así como las posibles mejoras a realizar sobre la aplicación.

- **Bibliografías y referencias**.

Listado de fuentes y documentos que han servido de referencia para la elaboración del proyecto.

2. Metodología

Cuando hablamos del desarrollo de software y de todos los procesos implicados en él, hablamos de algo complejo y difícil de manejar salvo que se tenga algún tipo de metodología. [3].

A día de hoy se puede distinguir entre metodologías predictivas y adaptativas. A continuación vamos a presentar las metodologías existentes para el desarrollo de software y se explicará cual ha sido la elegida y los motivos de la elección.

2.1 Metodologías predictivas

Las metodologías predictivas, desde el inicio, antes de comenzar a desarrollar el software, analizan el alcance del mismo de manera minuciosa y tratan de predecir con exactitud un coste y un tiempo en el que se deben desarrollar cada uno de los diferentes apartados en los que se ha considerado dividir el software.

La metodología predictiva más conocida es Métrica 3 [4], que es una metodología promovida por el Ministerio de Administraciones Públicas del Gobierno de España que ofrece a las Organizaciones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software.

El proceso de desarrollo de Métrica 3 contiene todas las actividades y tareas que se deben llevar a cabo para desarrollar un sistema, cubriendo desde el análisis de requisitos hasta la instalación del software, y consta de los siguientes procesos:

- *ESTUDIO DE VIABILIDAD DEL SISTEMA (EVS).* Se analiza un grupo concreto de necesidades para obtener una solución a corto plazo teniendo en cuenta aspectos técnicos, económicos, legales y operativos.
- *ANÁLISIS DEL SISTEMA DE INFORMACIÓN (ASI).* Se realiza una especificación detallada de los requisitos del sistema.
- *DISEÑO DEL SISTEMA DE INFORMACIÓN (DSI).* Se define la arquitectura del sistema y el entorno tecnológico en el que se va a realizar, así como los distintos componentes del sistema.

- *CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)*. Se realiza el desarrollo y prueba del sistema y se elaboran los manuales de usuario y explotación.
- *IMPLANTACIÓN Y ACEPTACIÓN DEL SISTEMA (IAS)*. En este último proceso, se realiza la implantación del sistema y en el caso de ser necesario se realiza también la formación del usuario para la utilización del sistema.

Este método es muy eficiente si tiene un diseño perfecto y no existen cambios posteriores a la realización de toda la documentación de los procesos, pero tener la certeza de que el diseño es perfecto no es algo que pueda realizarse de manera previa al desarrollo del sistema salvo en contadas ocasiones.

2.2 Metodologías adaptativas

Las metodologías adaptativas o desarrollo ágil, nacen como contrapartida a las metodologías predictivas. Este tipo de metodologías se basan en 4 valores y 12 principios que se desprenden del Manifiesto Ágil. [5]

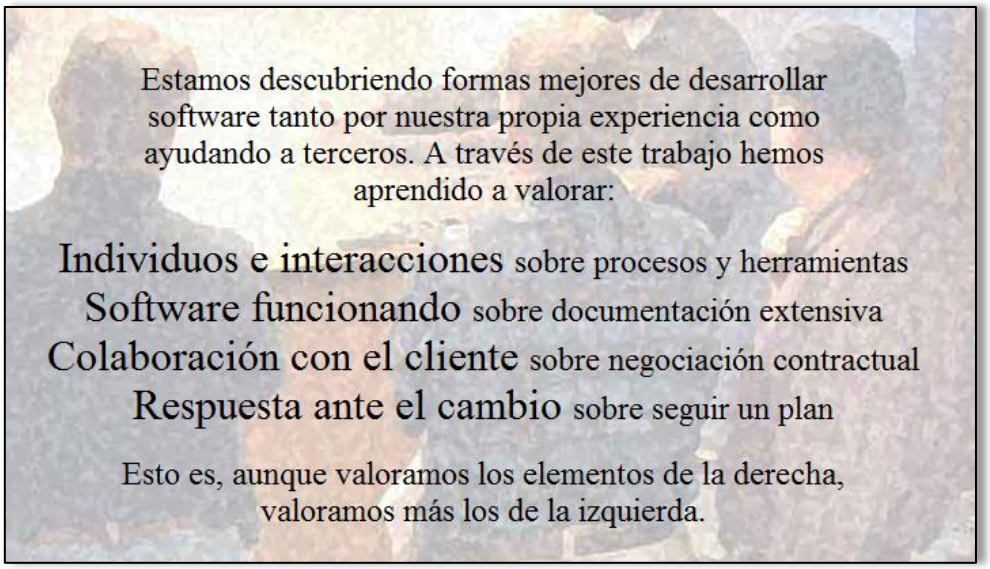


Ilustración 1. Valores del Manifiesto Ágil[3]

[3] http://agilemanifesto.org/iso/es

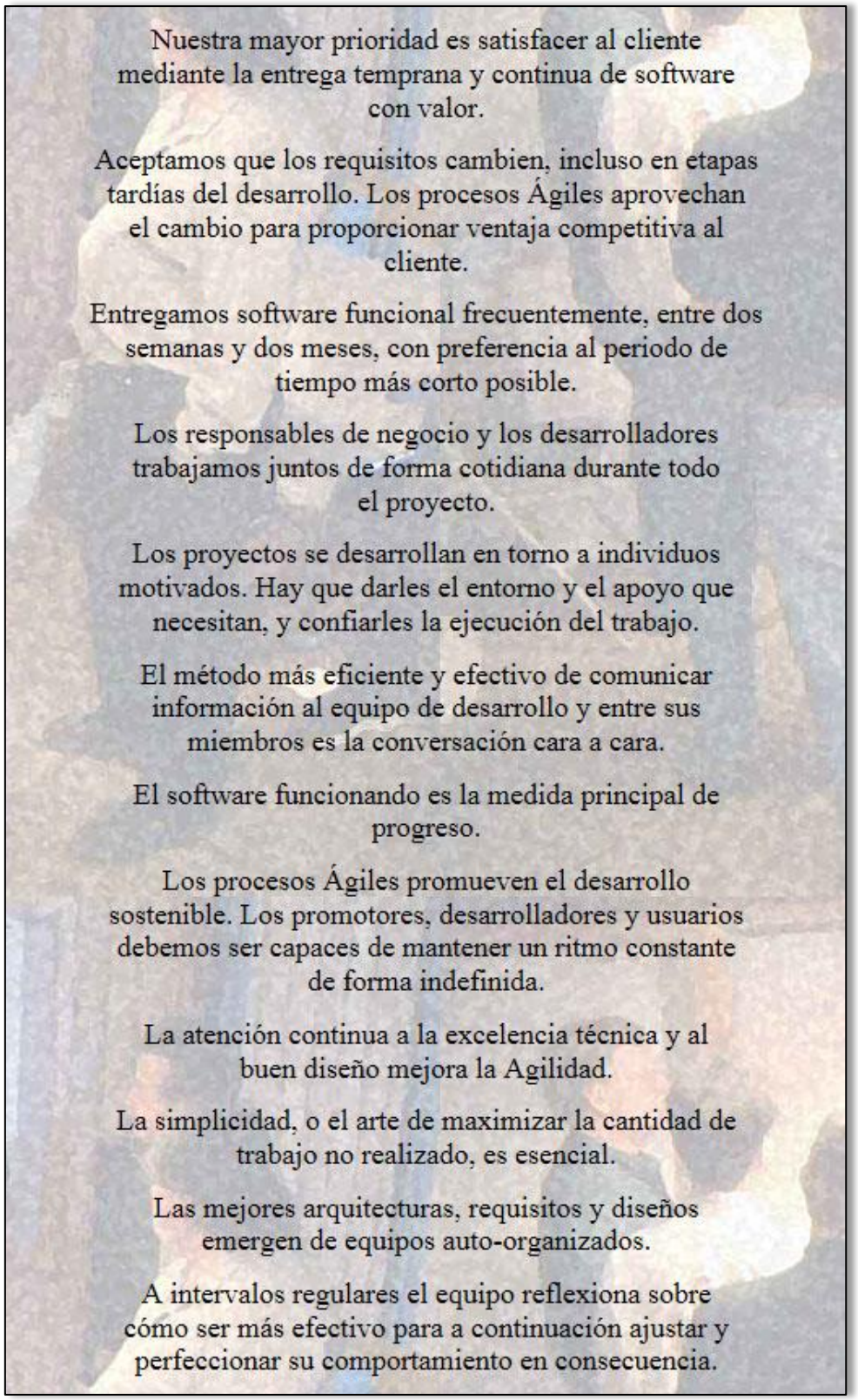


Ilustración 2. Principios del Manifiesto Ágil[4]

[4] http://agilemanifesto.org/iso/es/principles.html

En base a estos valores y principios, las metodologías adaptativas buscan una mayor interacción con el cliente, no excluyendo su participación en el proceso de desarrollo como se venía haciendo con las metodologías predictivas, en las cuales solo actuaba al principio en la toma de requisitos y al final con la prueba del producto.

Con estas nuevas metodologías, se busca que el cliente vaya teniendo una referencia constante de cómo va a ser el producto final, haciendo entregas periódicas (intervalos de entre 2 y 4 semanas, generalmente), y con ello también se evita que una vez terminado el producto y entregado al cliente, éste observe que lo que pidió en su momento no es realmente lo que quería o como lo quería.

Se reduce el volumen de documentación técnica, ya que aunque se realice una primera toma de requisitos en el inicio del proyecto, no se realiza una documentación exhaustiva y completa al detalle, ya que en cada entrega, el cliente puede ir indicando si seguir en la línea que se lleva o cambiar alguno de los aspectos que había predefinido anteriormente.

Hay que tener en cuenta que lo que se muestra al cliente en cada entrega, no se trata de un prototipo del funcionamiento del producto final, sino que se trata del propio producto final funcionando, lo que evita que entre el prototipo y el producto final haya diferencias que el cliente no admita.

Esos son los conceptos básicos de las metodologías adaptativas, pero luego dentro de cada metodología esos conceptos se pueden desarrollar de manera distinta. A continuación vamos a describir las más conocidas y utilizadas actualmente.

2.2.1 Crystal Clear

La metodología Crystal, engloba una familia de metodologías, que en función del tamaño del equipo, del volumen del proyecto y de la criticidad del mismo define metodologías diferentes. En este caso, Crystal Clear representa el nivel más bajo de criticidad y de número de integrantes del equipo.

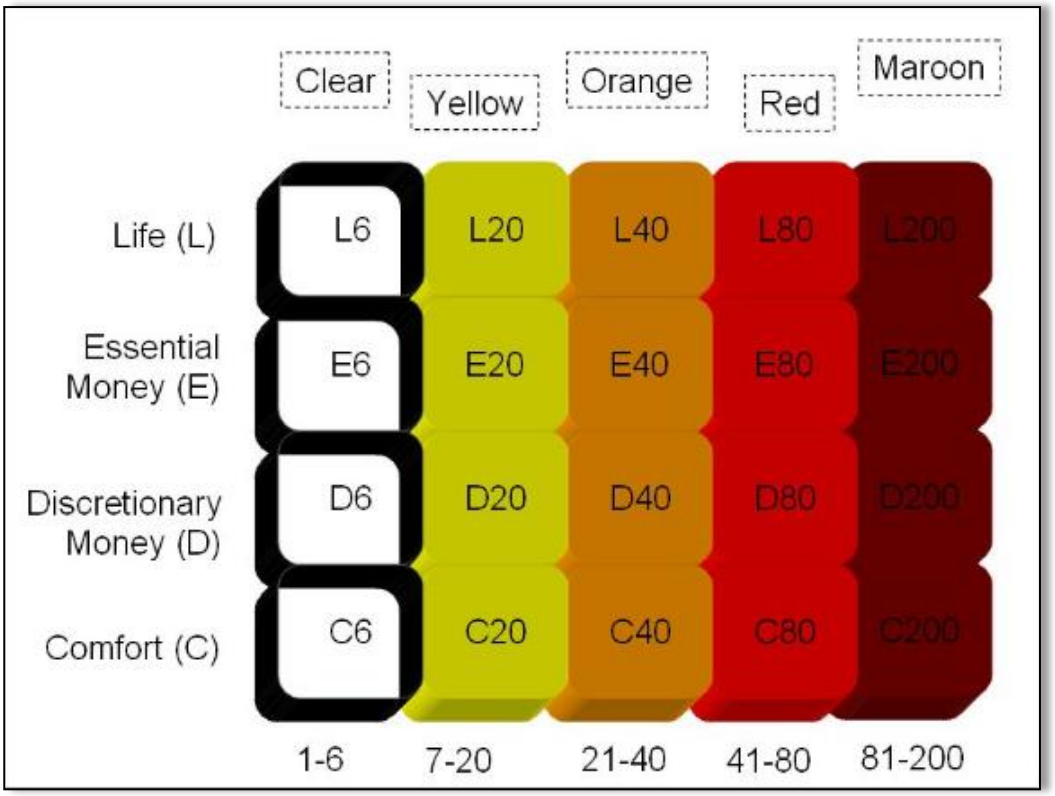


Ilustración 3. Metodologías Crystal

Las características de la metodología Crystal Clear son las siguientes:

- Alta frecuencia de entregas. Puede ser diaria, semanal o mensual.
- Espacio común de trabajo. Todos los integrantes del proyecto se deben encontrar en la misma sala.
- Mejora reflexiva. Reunión periódica para compartir opiniones y puntos de vista con el fin de mejorar el proyecto.
- Seguridad personal. Libertad de opinión sobre el desarrollo del proyecto.
- Enfoque. Definición clara de la tarea a realizar.
- Acceso a usuarios expertos. Localizados en la misma sala también.

2.2.2 SCRUM

Desarrollado por Ken Schwaber[5] y Jeff Sutherland[6], se basa en la entrega de partes del producto acabado en pequeños intervalos de tiempo de manera que en el caso de que haya cambios en los requerimientos una vez enseñado la parte del producto desarrollada, no se pierda una gran cantidad de tiempo.

Para ello se definen tres roles (Dueño del Producto, Scrum Master y Equipo) y una forma de trabajo en sprints (periodo de tiempo entre una entrega y otra) donde se sigue el siguiente procedimiento:

- El Dueño del Producto entrega al Scrum Master los requisitos del producto (Producto Backlog) con la prioridad con la que quiere verlos realizados.
- En la reunión de Sprint Planning, el equipo determina los elementos del Product Backlog que se comprometen a tener terminados en el Sprint, lo que conforma el Sprint Backlog. Este Sprint Backlog no se puede modificar a lo largo del Sprint.

Ilustración 4. Sprint Backlog

[5] https://en.wikipedia.org/wiki/Ken_Schwaber
[6] https://en.wikipedia.org/wiki/Jeff_Sutherland

- Dentro de cada Sprint, se realizan reuniones diarias del Equipo con el Scrum Master para revisar lo realizado y lo pendiente. Esto sirve para poder sincronizarse entre los miembros del Equipo y para detectar problemas y atajarlos lo más rápido posible.
- Una vez se ha terminado el Sprint se realiza una revisión del mismo para ver lo que se ha completado y lo que no del Sprint, y se muestra lo realizado al Dueño del Producto.
- Después de la revisión, se realiza por último, una retrospectiva del Sprint en el que todo el equipo interviene comentando los detalles del Sprint para poder mejorar en los siguientes Sprints y/o proyectos.

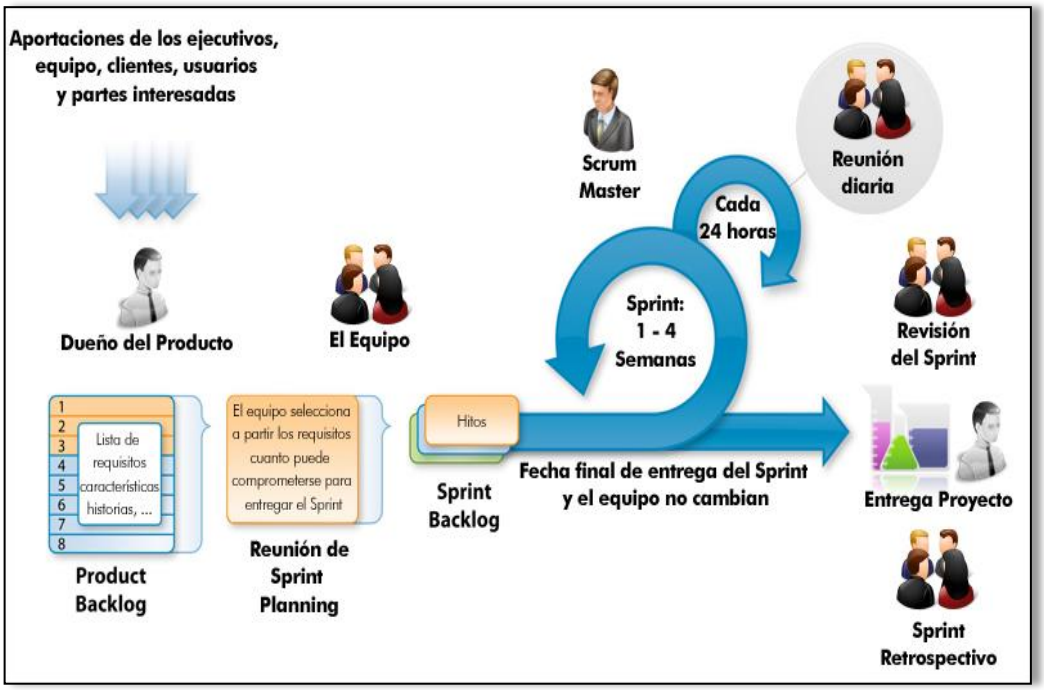


Ilustración 5. Metodología SCRUM

2.2.3 Método de desarrollo de sistemas dinámicos (DSDM)

Creado en 1995, actualmente se encuentra en su versión 4.2 y tiene como principios básicos los siguientes [6]:

- Implicación activa de los usuarios.
- Autonomía de los desarrolladores para tomar decisiones.

- Alta frecuencia de entregas incrementales.
- Cambios reversibles.
- Las pruebas forman parte de la fase de desarrollo.
- Actitud colaborativa entre todos los integrantes del proyecto.
- Requisitos establecidos a nivel general.

La siguiente ilustración muestra el ciclo de vida de DSDM, donde se puede observar una iteración constante de un entregable con iteraciones dentro de cada fase.

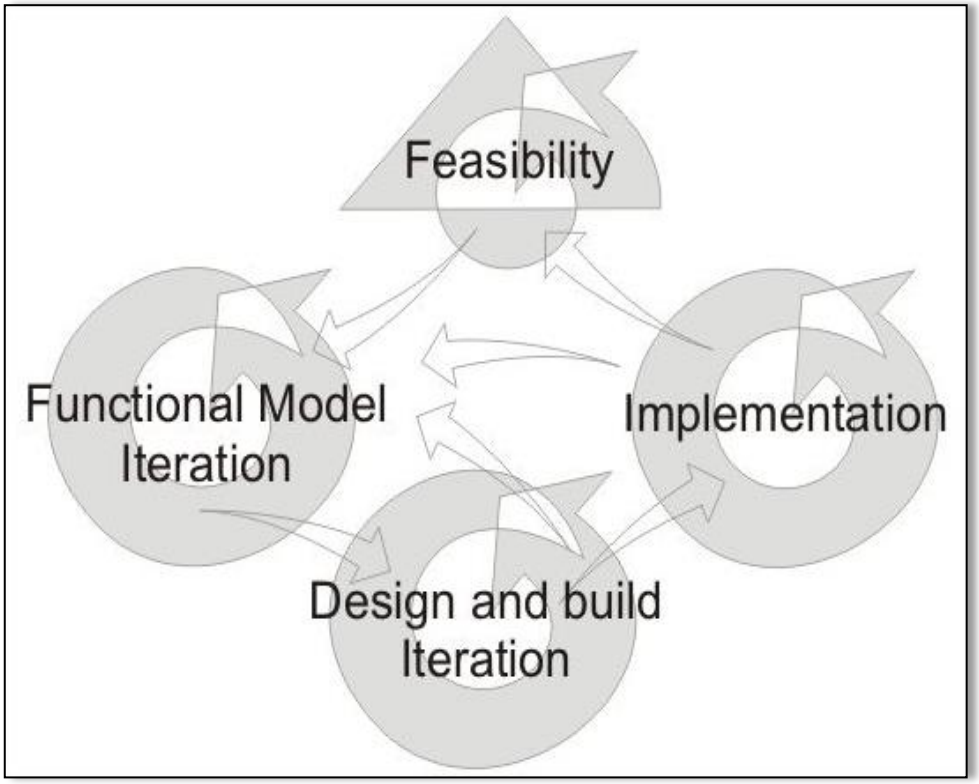


Ilustración 6. Metodología DSDM

2.2.4 Lean Software Development

La metodología Lean, desarrollada por Mary y Tom Poppendieck, es una adaptación de la metodología de trabajo de la fábrica de Toyota, que se basa en el modelo de producción "just in time"[7].

[7] https://es.wikipedia.org/wiki/M%C3%A9todo_justo_a_tiempo

La idea de esta metodología es construir solo lo necesario eliminando todo lo que no aporte valor, y en el caso de que algo no vaya bien parar. Los principios básicos de esta metodología son:

- Eliminar desperdicios (burocracia, código innecesario…)
- Calidad continua (no esperar al final del producto para aplicar la calidad)
- Formación continua del equipo
- Decidir lo más tarde posible (decisiones que sean irreversibles o casi)
- Entregas rápidas.
- Potenciación del equipo (el equipo debe trabajar cómodo y con confianza en sus tareas)
- Ver el conjunto (tener claro el objetivo general en el que se incluye la tarea particular)

2.2.5 Extreme Programming (XP)

Desarrollada por Kent Beck[8], posee muchas características en común con SCRUM y Crystal Clear, con la diferencia de que le da extrema importancia a la calidad del código que se exige como manera de producir de manera sostenible [7]

Al igual que el resto de metodologías ágiles, la idea de XP es la adaptación a los cambios de requisitos en cualquier punto del ciclo de vida del proyecto. Para ello, son necesarias las siguientes prácticas básicas:

- Integración del equipo. El equipo se compone de toda la gente involucrada en el proyecto, desde el cliente hasta los desarrolladores.
- Pruebas continuas y automáticas propuestas tanto por cliente como por desarrolladores
- Planificación continua. La planificación de los hitos se revisa constantemente.
- Código estandarizado, simple y compartido. Todos los desarrolladores conocen el código completo, y este ha de crearse de

[8] https://es.wikipedia.org/wiki/Kent_Beck

la manera más sencilla posible y con un estándar de codificación que permita la actuación de cualquier desarrollador en cualquier parte del código.

- Carga de trabajo continua. Se deben evitar los sobreesfuerzos y los valles de trabajo.
- Entregas constantes (intervalos de entre 1 y 3 semanas).
- Refactorización constante de código.
- Programación en parejas.
- Corrección de todos los errores antes de incluir nada nuevo.

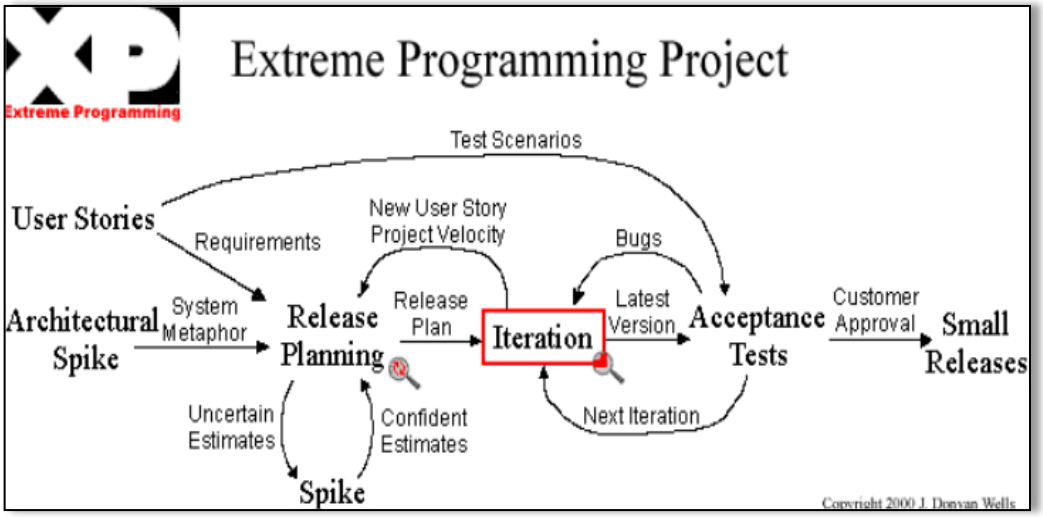


Ilustración 7. Metodología XP

2.2.6 Adaptive Software Development

Desarrollada por Jim Highsmith[9] y Sam Bayer, está basada en la imposibilidad de predecir y medir el desarrollo, y propone la adaptación continua a circunstancias cambiantes.

Para ello define tres etapas, iterativas como el resto de metodologías ágiles, que son:

[9] https://en.wikipedia.org/wiki/Jim_Highsmith

- Especular. En esta primera etapa, debemos definir los principales objetivos y metas del proyecto, teniendo en cuenta que pueden sufrir variaciones.
- Colaborar. A través de la coordinación e implicación de todo el equipo, se realiza el desarrollo del trabajo.
- Aprender. Se recapitula todo lo realizado hasta el momento y se tienen en cuenta a la hora de realizar la siguiente especulación.

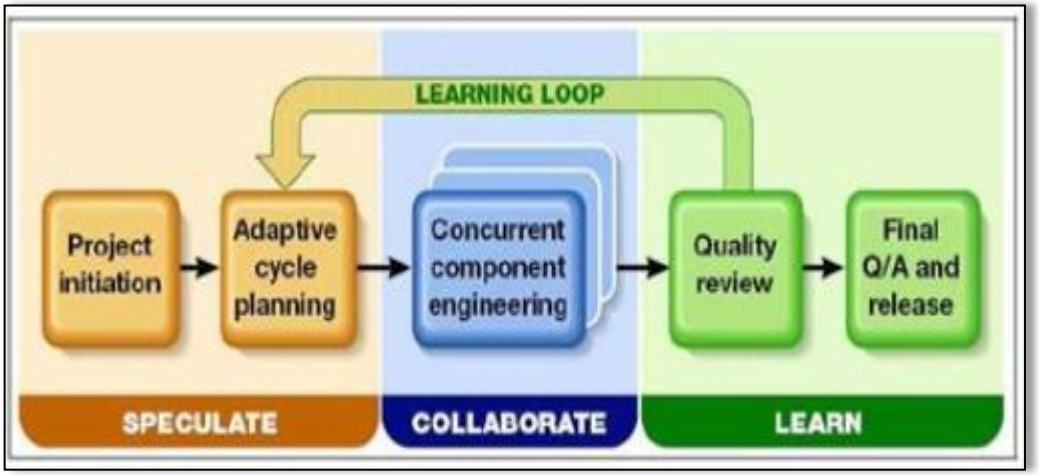


Ilustración 8. Metodología ASD

2.3 Elección de la Metodología

Una vez conocidas las principales metodologías ágiles, se puede concluir que al partir todas de los mismos principios (manifiesto ágil), todas presentan una estructura similar (iteraciones, entregas, revisión) con pequeños matices que pueden ir basados en función de motivos como el tamaño del proyecto, la urgencia de entrega, la frecuencia de los entregables, etc…

Las metodologías más usadas actualmente son SCRUM y XP, cuando no una mezcla de las dos, ya que una está más centrada en la gestión (SCRUM) y la otra está más centrada en el desarrollo (XP). De hecho, realizando una búsqueda en internet de ambas

palabras, se puede observar que en multitud de sitios aparecen complementándose la una a la otra.

Realizando una pequeña comparativa entre ambas metodologías sobre un par de aspectos básicos, obtenemos el siguiente cuadro:

	SCRUM	XP
Requisitos	Fijos	Modificables
Iteraciones	Fijas, de 2 a 4 semanas	Variables, 1 a 3 semanas
Interacción de los miembros del proyecto	Producto Owner y el Equipo no tienen por qué tener relación directa	El equipo desde el cliente al desarrollador, debe estar integrado
Enfoque	Gestión	Desarrollo
Tamaño del proyecto	Mediano-Grande	Pequeño-Mediano

Teniendo en cuenta las características de nuestro proyecto, a saber, proyecto pequeño con unos requisitos finales que no están definidos al 100%, en el que nos interesa hacer entregas cada poco tiempo para ver la evolución del mismo y ver si seguir por ese camino o no, en el que prima es el desarrollo de la aplicación por encima de la gestión del mismo, y en el cual el equipo está en continuo contacto para evaluar el camino a seguir, nos decidimos por seguir la metodología de programación extrema (XP) por ser la que mejor se adapta tanto a nuestras características como a las del proyecto (salvo por la programación en parejas, que por motivos obvios no se puede realizar)

3. Estado del Arte

En este apartado, se realizó un estudio de algunas de las distintas aplicaciones que existen actualmente para la creación de avatares, así como una explicación del motor de juegos en el que buscamos integrar el avatar resultante de nuestra aplicación.

Abordaremos la importancia de los avatares hoy en día, y realizaremos una comparativa entre distintas aplicaciones de personalización de avatares que podemos encontrar navegando por la red.

Realizaremos por último el estudio sobre tres aplicaciones, cada una de las cuales se centra en un determinado aspecto del avatar, de manera que podemos obtener datos acerca de lo que queremos que sea nuestro resultado final para desarrollar un avatar completo.

3.1 Personalización de avatares

Los avatares personalizados son uno de los rasgos que definen a los mundos virtuales y se han convertido en una característica importante en los mismos. En los juegos sociales, ocurre lo mismo, ya que cada vez más, la sociedad demanda información actualizada al instante, ya sea acerca de una noticia internacional o de un suceso personal de algún amigo. Es por eso que es necesaria una aplicación con la que actualizar el avatar creado en una primera instancia en el ingreso en un foro, juego social, mundo virtual, etc…

Al principio, el avatar se creaba una vez ingresabas en alguno de esos sitios, y una vez creado, no se sentía la necesidad de modificarlo, por lo que a pesar de existir aplicaciones para crear avatares virtuales, esta función no era una preocupación esencial ni para el usuario ni para el responsable del sitio.

Pero como ya se ha comentado, esta idea ha cambiado, y cada vez son más el número de aplicaciones que surgen para personalizar avatares, ya sea para utilizarlos en foros, blogs o directamente para integrarlos en juegos ya existentes, y así satisfacer esa nueva necesidad de actualización casi inmediata que requieren los usuarios.

3.2 Aplicaciones de personalización de avatares en la red

Existen múltiples aplicaciones de creación de avatares, cada una con sus características propias y cada una diferente al resto por el tipo de avatar que se quiere crear y la utilización que de él se quiere hacer.

Dependiendo del tipo de utilización que se le quiera dar, tenemos programas que nos permitirán crear avatares siguiendo un estilo de dibujo (crear avatares con estilo manga[10]) o en base a un modelo característico de avatar del cual solo nos permitirán cambiar pequeños aspectos del avatar como colores de las características predefinidas (crear avatares con el estilo de un muñeco de South Park[11]), y aplicaciones en las cuales podemos personalizar hasta el más mínimo detalle de un complemento[12].

Ilustración 9. Tipos de Avatares

Dentro de esas tipificaciones de programas, existen desde aplicaciones en las que el avatar se limita al rostro del personaje, hasta aplicaciones que permiten, no solo definir el cuerpo entero, sino también dotarle de complementos y accesorios de distinto tipo.

[10] http://www.faceyourmanga.com
[11] http://www.southparkstudios.com/avatar/
[12] http://www.doppelme.com

A continuación, se muestra una tabla comparativa sobre algunas de las distintas aplicaciones que podemos encontrar navegando por la web, en la que identificaremos la aplicación examinada, el estilo de dibujo que muestra el avatar, una breve descripción del avatar creado y el nivel de detalle que alcanza en las características de la cara, la ropa y los accesorios del avatar.

Aplicación	Estilo	Descripción	Nivel de detalle		
			Cara	Ropa	Accesorios
Anime Avatar Creator [8]	Anime	Permite crear un avatar solo de representación facial de estilo anime.	Alto	No	Alto
Avatar Simulator for Gaia [9]	Fantástico	Permite crear un avatar para la red social Gaia	Medio	Alto	Alto
Create Your Own Superhero [10]	Superhéroe	Permite crear un avatar de tu propio superhéroe	Medio	Alto	Alto
Doppel Me [11]	Infantil	Permite crear un avatar de cuerpo entero con un dibujo infantil	Bajo	Alto	Alto
Face Your Manga [12]	Manga	Permite crear un avatar de medio cuerpo con especial atención en la cara de estilo manga.	Alto	Medio	Medio
My Avatar Editor [13]	Mii	Permite crear un avatar al estilo de los caracteres Mii de la consola de videojuegos Wii de Nintendo.	Alto	Bajo	No
Pick a Face [14]	Realista	Permite crear un avatar de medio cuerpo de estilo realista	Alto	Bajo	Bajo
Portrait Illustration Maker [15]	Realista	Permite crear un avatar de medio cuerpo con un estilo realista.	Alto	Bajo	Bajo
Scott Pilgrim avatar creator [16]	Scott Pilgrim	Permite crear un avatar con la apariencia de un personaje de la película Scott Pilgrim contra el mundo.	Alto	Alto	Alto
South Park avatar creator [17]	South Park	Permite crear un avatar con la forma y características de un muñeco de South Park	Bajo	Alto	Alto

Ilustración 10. Tabla comparativa de aplicaciones

Ante el abanico de diferentes tipos de aplicaciones existentes, se ha realizado el estudio de tres de ellas teniendo en cuenta que no se quería hacer un avatar con el estilo de ningún dibujo conocido, y que buscábamos realizar una aplicación que pudiese definir un avatar con el suficiente detalle para que el usuario quedase contento, de manera que pudiese decidir sobre todos los aspectos del avatar.

Las tres aplicaciones elegidas para el estudio han sido: Doppel Me, Portrait Illustration Maker y My Avatar Editor.

3.2.1. Doppel Me

Esta aplicación nos presenta un avatar con una posición y forma fijas, cuya primera opción antes de continuar con la modificación del avatar es la definición de género y la elección de color de piel y color de ojos.

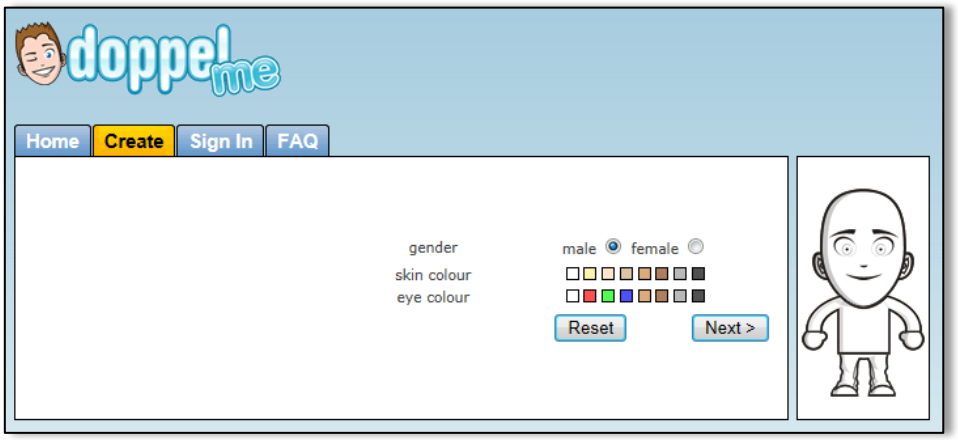


Ilustración 11. Pantalla de Inicio DoppelMe

La diferencia entre elegir un género u otro, no es otra que el color de labios, ya que mientras si se elige el género masculino la boca presenta un trazo de color negro, cuando se selecciona el género femenino, los labios presentan un trazo de color rojo.

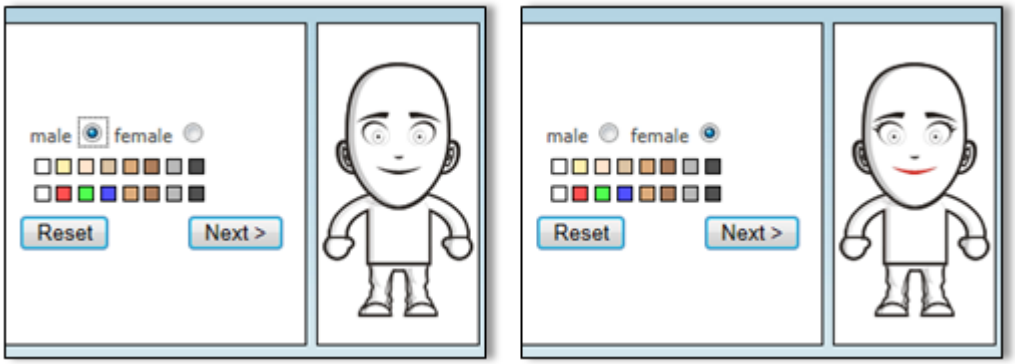


Ilustración 12. Pantalla de selección de género DoppelMe

Una vez se continúa después de la selección del género y el color de piel, la aplicación muestra tres zonas diferenciadas en las cuales se muestra:

- el tipo de caracteristica que se quiere modificar (zona lateral izquierda);
- las distintas opciones que ofrece cada caracteristica, con la posibilidad de mostrar mas opciones a traves de flechas de desplazamiento (zona central);
- los resultados de las distintas modificaciones que se van realizando sobre el avatar

Ilustración 13. Pantalla de características y opciones de selección DoppelMe

En la zona central también muestra la opción de volver a la pantalla de inicio para poder resetear la imagen y/o volver a modificar el género, color de piel y color de ojos, y dar el avatar por concluido, con lo que se nos abre otra pantalla en la que nos solicitara una dirección para el envío y se nos invitara a registrarnos para poder acceder a algunas opciones de características que en la pantalla anterior nos aparecen como bloqueadas al no estar registrados (opciones que aparecen sombreadas en la imagen de arriba).

Ilustración 14. Pantalla de grabación DoppelMe

Como se puede observar esta aplicación está básicamente dedicada a la vestimenta del avatar y sus complementos, ya que dentro de las características físicas, la única que permite modificar es la expresión de la cara, siendo toda ella un conjunto y no pudiendo elegir entre la mezcla de varias separadas.

3.2.2. Portrait Illustration Maker

Esta aplicación se focaliza en la customización del tercio superior del avatar, y esta principalmente diseñada para ser utilizada en foros y chats, en los que suele aparece una pequeña imagen junto a los comentarios realizados por el usuario.

Al entrar en la web, se nos presenta el contorno de una cara vacía sin fondo ni ningún tipo de característica. También se nos presentan todas las características modificables en la parte superior de la pantalla.

Para hacernos a una idea de lo que la aplicación puede hacer, se nos presentan algunas imágenes de muestra y como pueden ser utilizadas. También se nos presenta la posibilidad de generar un avatar aleatorio por si no queremos estar viendo todas las posibilidades de la aplicación.

Ilustración 15. Pantalla de Inicio Portrait Illustration Maker

Una vez seleccionamos alguna característica, nos aparecen un gran número de opciones dentro en la parte inferior de la pantalla con un *scroll* horizontal para poder verlas todas. En cada opción muestra el contorno del avatar con la posición exacta de la opción dentro del avatar para poder observar cómo quedaría su resultado.

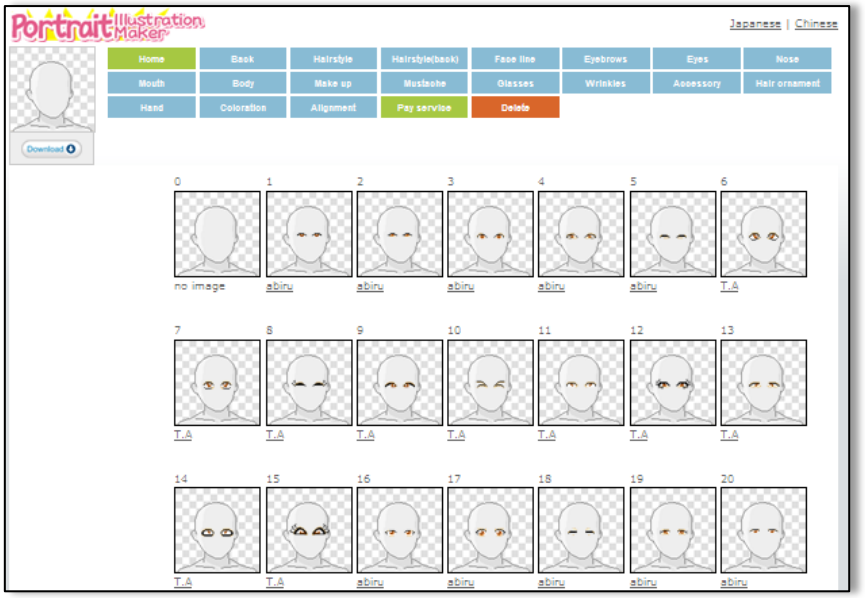


Ilustración 16. Pantalla de opciones de Portrait Illustration Maker

Cada vez que se selecciona una opción, esta se aplica en el avatar que estamos creando, que se encuentra situado en la parte superior izquierda de la pantalla.

Ilustración 17. Visualización de avatar Portrait Illustration Maker

Una vez se han realizado todas las modificaciones en las características, del avatar, esta aplicación permite el cambio de color de alguna de sus características en la pestaña “Coloration”. Permite la modificación del color del pelo, la piel, los ojos, los labios y la ropa que aparece en el avatar si es que hemos seleccionado alguna.

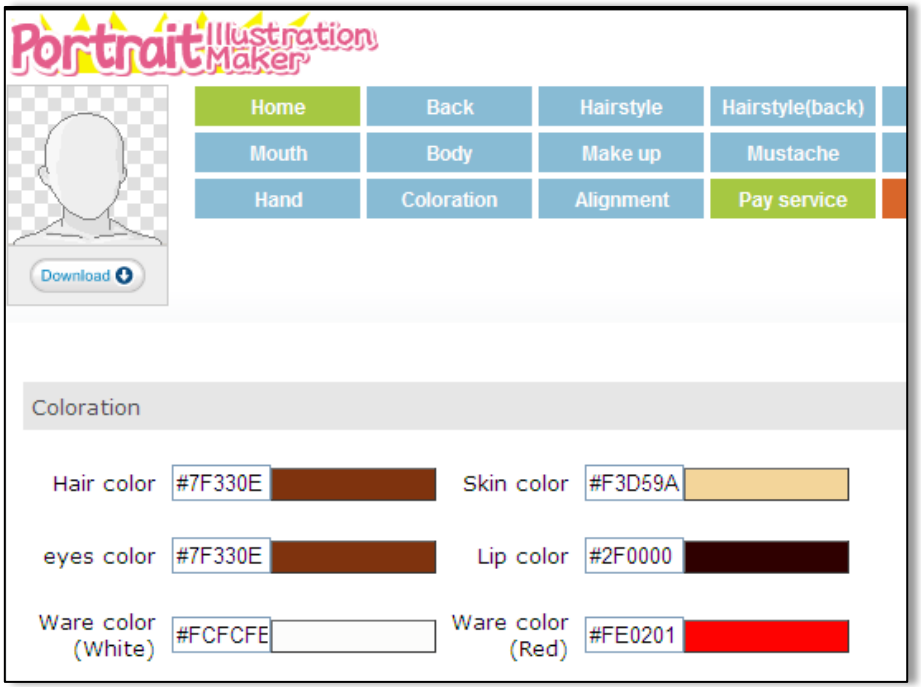


Ilustración 18. Selección de colores Portrait Illustration Maker

También permite la colocación de cada característica donde nosotros queramos, dando la oportunidad de desplazar cada característica tanto horizontal como verticalmente.

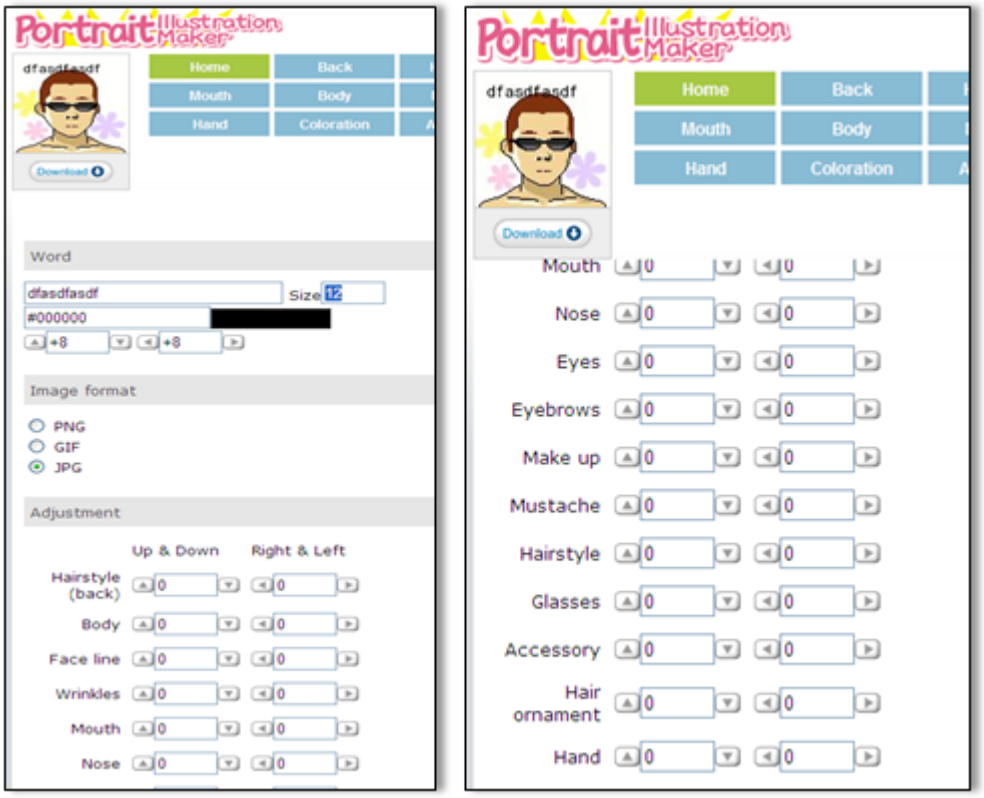


Ilustración 19. Pantalla de recolocación de características Portrait Illustration Maker

En el momento en el que queramos guardar nuestro avatar, lo único que se debe hacer es pulsar en la opción "*Download*" que aparece justo al pie del avatar que nos va mostrando los cambios que realizamos y se nos abrirá un cuadro de dialogo para descargar la imagen a nuestro soporte.

3.2.3. My Avatar Editor

Este editor de avatares se asemeja en su resultado final a los avatares creados para la consola Wii de Nintendo.

De las tres aplicaciones analizadas, quizá es esta la que mayores posibilidades ofrece, ya que aúna la versatilidad de Portrait Illustration Maker en cuanto a las diferentes opciones de cada característica, con la visión de cuerpo entero del avatar que nos ofrece Doppel Me.

La aplicación nos mostrará siempre al pie del panel, las diferentes características que podremos modificar, así como la opción de descargar el avatar a algún tipo de soporte. El avatar aparecerá siempre a la izquierda del panel con las modificaciones que le vayamos realizando, y a su derecha se encontraran las diferentes opciones de modificación para cada característica.

Para comenzar, esta aplicación nos ofrece la posibilidad de elegir el género del avatar, así como su color favorito (que será el color destinado a su camiseta) y su fecha de cumpleaños. También nos da la opción de generar un avatar aleatorio por si no queremos estar configurando característica a característica.

Ilustración 20. Pantalla de Inicio My Avatar Editor

Esta aplicación, a diferencia de las otras, permite personalizar la complexión física del avatar, pudiendo elegir la altura y anchura del mismo a través de dos barras de desplazamiento que nos permiten definir el avatar a nuestro gusto.

Ilustración 21. Cambio de complexión My Avatar Editor

Cuando seleccionamos una característica para modificar, la aplicación realiza un zoom sobre la zona a modificar para poder apreciar mejor los cambios realizados.

Dentro de cada característica, la aplicación nos ofrece todas las opciones posibles de modificación en bloques de nueve elementos. Si alguna característica posee más de 9 opciones posibles, nos ofrecerá la posibilidad de recorrer el resto de opciones a través de flechas de desplazamiento indicándonos en cada momento en que bloque nos encontramos y de cuantos bloques de opciones consta la característica.

Ilustración 22. Pantalla de opciones My Avatar Editor

Aparte de poder elegir la opción que queramos, la aplicación nos da la posibilidad de cambiar el color de la casi todas las características en las que nos encontremos a través de unos botones de colores predefinidos que no pueden ser cambiados. También permite otras modificaciones tales como el cambio de tamaño, la distancia de cada opción con respecto al eje, el ángulo de la característica y su posición, tanto horizontal como vertical.

Ilustración 23. Pantalla de recolocación de características My Avatar Editor

A la hora de guardar el avatar, la aplicación nos ofrece varias posibilidades.

- podemos guardar el avatar en el formato de la aplicación para posteriormente si queremos, volver a cargarlo y modificarlo según se quiera;

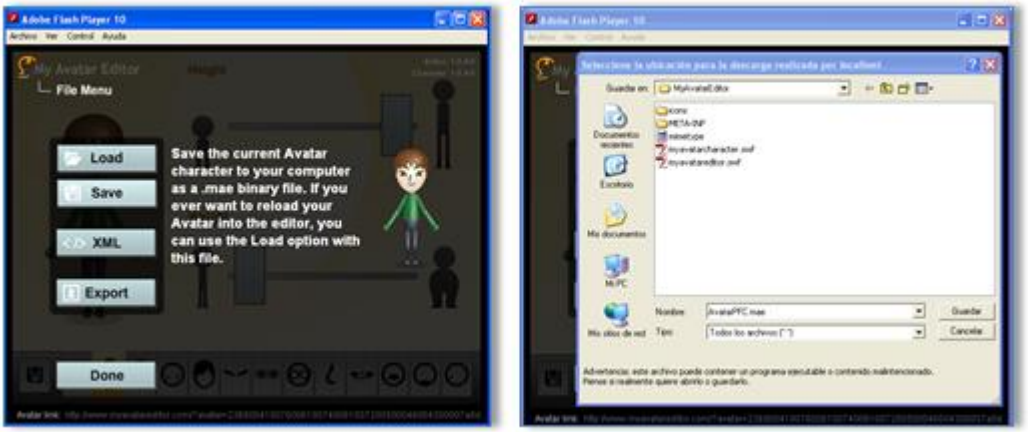


Ilustración 24. Pantallas de guardado My Avatar Editor (I)

- podemos guardar simplemente la imagen en formato jpg o png con la posibilidad de quitarle la sobra de avatar, de guardar solo la cara del avatar, o de cambiar o quitar el color de fondo de la imagen;

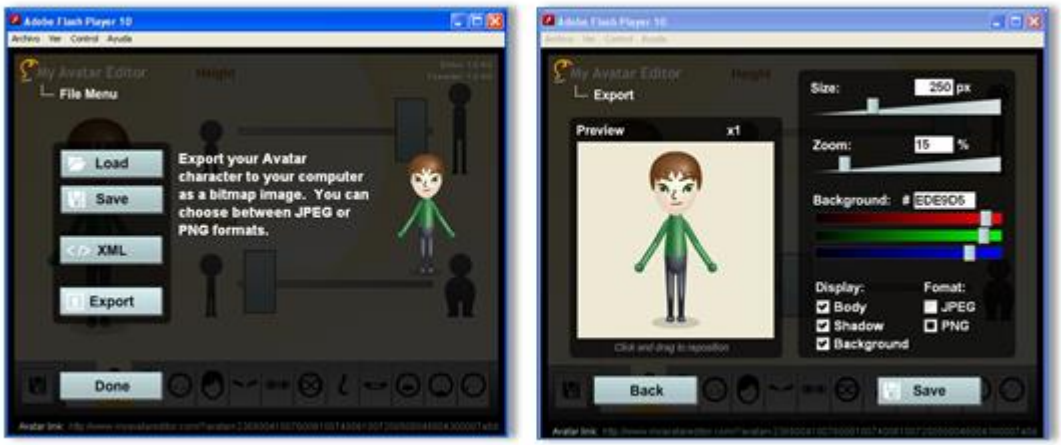


Ilustración 25.Pantallas de guardado My Avatar Editor (II)

- podemos modificar el xml manualmente y observar su efecto en el avatar

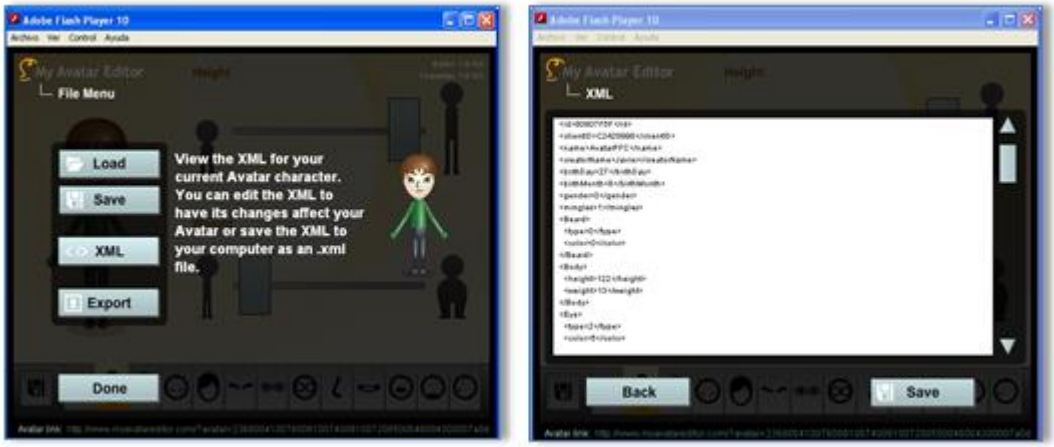


Ilustración 26. Pantallas de guardado My Avatar Editor (III)

3.3 Motor OpenSpace

OpenSpace es un motor de mundos virtuales isométricos (2D con perspectiva en 3D) basado en Flash[13], que "*aprovecha la potencia de ActionScript 3*[14] *y SmartFoxServer"* [18] para dar la posibilidad al diseñador de juegos y mundos virtuales de especificar un alto nivel de características de este tipo de entornos habitualmente no abarcadas por motores similares, como por ejemplo, la posibilidad de modificar los mapas en tiempo de ejecución, permitiendo al usuario añadir o eliminar elementos a los mismos. Además, OpenSpace permite la creación de arquitecturas de mapas avanzados, con la posibilidad de crear puentes y cambios de nivel, manejando el fondo del mapa de manera independiente al avatar, lo que permite un mejor rendimiento de la representación.

OpenSpace basa los mapas en una estructura de baldosas, que permite a los avatares moverse y no traspasar los elementos creados en el mapa, así como también permite realizar zoom sobre un mapa y navegar por el mismo sin necesidad de tener que dirigir el avatar a la zona que queramos ver, lo que evita que para ver un mapa tengamos que recorrerlo completamente con el avatar. El propio OpenSpace es el responsable del algoritmo de movimiento del avatar, utilizando un algoritmo de búsqueda en anchura (BFS)[15], que hace recorre al avatar el camino más corto posible entre dos puntos.

Del lado del desarrollador, OpenSpace deja la responsabilidad del aspecto y las direcciones del movimiento del avatar (4 u 8 direcciones), así como la posibilidad de configurar alguna acción predeterminada en caso de que el avatar se sitúe en un punto concreto del mapa gracias a un sistema avanzado de eventos que puede ser configurado por el desarrollador.

En definitiva, OpenSpace nos ofrece la posibilidad de, a partir de un avatar diseñado por el usuario, navegar por distintos mapas e interactuar con los objetos que en ellos encontramos.

[13] http://www.adobe.com/es/flashplatform/
[14] http://www.adobe.com/devnet/actionscript.html
[15] http://es.wikipedia.org/wiki/B%C3%BAsqueda_en_anchura

4. Análisis

Tenemos que tener en cuenta que el proyecto no sólo debe constar de la creación del módulo de personalización de avatares, sino que también debe abordar la correcta integración de los avatares creados en el entorno del servidor donde se va a desarrollar el juego, que independientemente del tipo de juego que sea, habrá de dibujar correctamente los avatares confeccionados por los usuarios en el mundo virtual que se haya creado para la ocasión.

Por lo tanto, no solo se tendrá que crear una aplicación capaz de configurar un avatar al gusto del usuario, sino que además, se tendrán que desarrollar las herramientas necesarias para la correcta integración del avatar en el juego.

4.1 El avatar

El módulo de creación de avatares debe proporcionar al usuario la capacidad de crear un avatar desde cero y poder almacenar dicha creación de manera que posteriormente pueda ser utilizada en el motor de juegos.

Partiendo de la premisa de que cada usuario puede decidir cambiar el aspecto de su avatar en cualquier momento, la aplicación debe estar dotada de la posibilidad de modificación de un avatar creado anteriormente, por lo que debe ser capaz de recuperar un avatar creado anteriormente a fin de poder realizar dichas modificaciones.

También hay que tener en cuenta que la aplicación debe poder facilitar el acceso a los usuarios de manera única y exclusiva, de modo que cada usuario acceda solamente a su avatar creado, y no pueda modificar o alterar los avatares creados por otros usuarios.

Para ello, la aplicación debe estar dotada de una pantalla de entrada con verificación mediante un usuario y contraseña que solo podrán ser conocidos por el

propio usuario, de manera que pueda proporcionar al usuario la seguridad, de que única y exclusivamente él, pueda acceder a modificar su avatar.

En el caso de que se trate de un usuario nuevo, la aplicación proporcionara la funcionalidad de crear un nuevo usuario, con la condición de que el nombre utilizado para identificar el usuario no exista ya, validándolo contra la base de datos de la aplicación.

Ilustración 27. Acceso aplicación

Para los casos de usuarios nuevos, la aplicación comenzará mostrando un avatar predefinido, mientras que para usuarios ya registrados, se mostrará el avatar guardado en el último acceso a la aplicación. El sistema se encargará de recuperar y cargar en pantalla el avatar correspondiente al usuario.

Una vez en la aplicación, el usuario deberá poder definir su propio avatar, por lo que la aplicación deberá ofrecerle la posibilidad de modificar, dentro de un abanico de características físicas, entre distintas opciones a fin de poder configurar el avatar a su gusto.

Dado el objetivo final de público infantil con el que va a realizarse la aplicación, ésta debe estar dotada de una interfaz gráfica sencilla y fácilmente comprensible, de

manera que los pasos a realizar sean intuitivos y no precise de un complejo manual de usuario.

Con este motivo, la aplicación constará de una pantalla única, en la que siempre se podrá observar el nombre del usuario, las características objeto de modificación y las opciones de almacenamiento y salida de la aplicación.

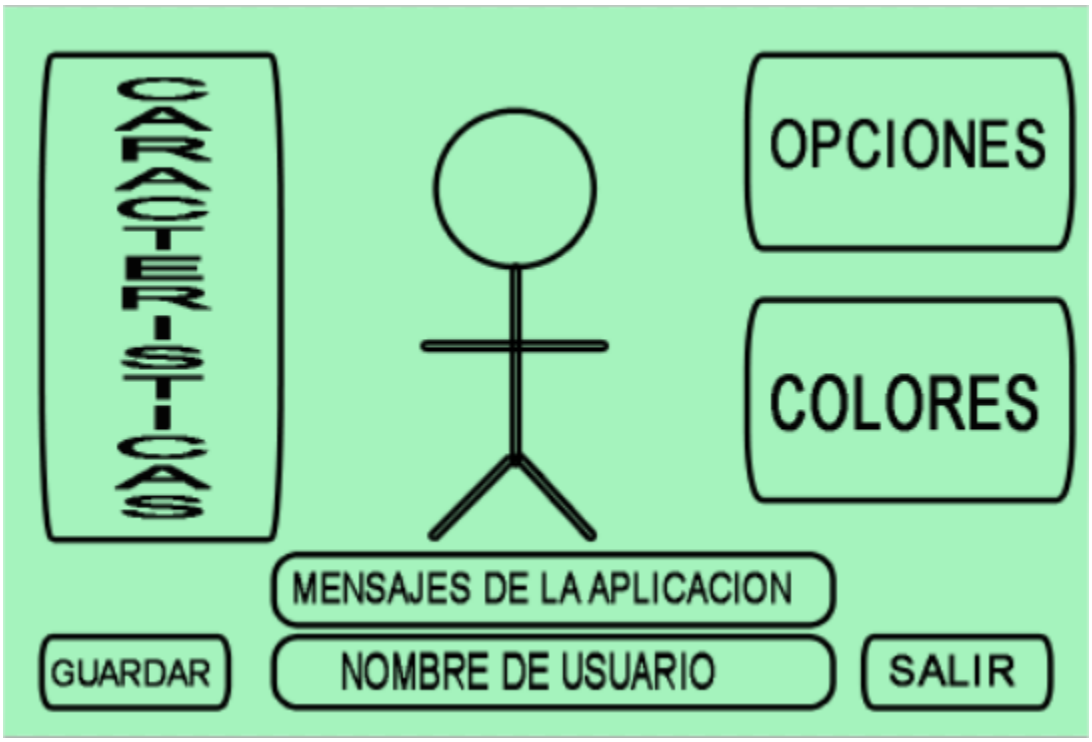


Ilustración 28. Esquema de la pantalla principal

En función de la distinta característica que se pulse para modificar, aparecerán para cada característica, las distintas opciones posibles de modificación, así como la posibilidad de cambiar el color de la característica si ello es posible.

Las características que vamos a definir como criterios básicos, se van a centrar en la definición de la cara del avatar, dejando en un segundo plano la vestimenta del mismo.

Como requisitos básicos deberíamos contar con las siguientes características:

- **Cara**: Definirá la forma geométrica de la cara con la posibilidad de modificar la tonalidad de la piel.
- **Pelo**: Definirá el corte de pelo del avatar así como el color del mismo que desea
- **Ojos**: Definirá la forma y color de los ojos.
- **Nariz**: Definirá la forma de la nariz.
- **Boca**: Definirá la forma de la boca.
- **Orejas**: Definirá la forma de las orejas.
- **Camiseta**: Definirá el tipo de camiseta así como el color del que la desea.
- **Pantalón**: Definirá el tipo de pantalón/falda que desea, con la posibilidad de variar el color.
- **Zapatillas**: Definirá el tipo de calzado del avatar y su color.

Dentro de cada característica se le debe dar la posibilidad de elegir entre distintas opciones, que aparecerán situadas en la parte superior derecha de la pantalla, y justo debajo de esas opciones, en caso de que sea posible, se le proporcionara al usuario una gama de colores de entre los cuales elegir para la opción seleccionada.

Una vez realizadas todas las modificaciones deseadas por el usuario (o en el momento que desee durante las modificaciones), la aplicación le permitirá grabar los cambios, de manera que ante cualquier problema del equipo, no pierda los cambios realizados hasta el momento.

Así mismo, y durante cualquier momento del proceso, la aplicación dará la posibilidad al usuario de salir de la misma sin realizar cambio alguno.

4.2 Integración en el motor de juegos

Una vez tenemos definido cuales son los requisitos necesarios para tener lista nuestra aplicación de creación de avatares, debemos ahora pasar a definir los requisitos de integración en el motor de juegos.

Debemos poder integrar el avatar creado en el motor de juegos para poder utilizarlo en el contexto en el que se incluya el juego.

Para ello, dotaremos al avatar del movimiento necesario para poder recrear el avatar creado por cada usuario dentro del espacio virtual que hayamos seleccionado.

La integración ha de ser posible realizarla en cualquier juego incluido en el motor de openspace, no limitándose a un único escenario.

4.3 Identificación de requisitos

En este apartado vamos a enumerar los requisitos identificados para la construcción de la aplicación. Estos requisitos, van a ser la parte principal de cada iteración dentro de la metodología XP, ya que cada *User Storie* (US a partir de ahora), va a dar lugar a uno o varios requisitos del sistema (REQ a partir de ahora).

US1 – Crear Usuario

REQ1.1	
Nombre	Solicitar usuario y contraseña
Descripción	La aplicación debe mostrar una pantalla en la que se le solicite al usuario, el nombre de usuario y contraseña que desea.

REQ1.2	
Nombre	Verificar usuario y contraseña
Descripción	La aplicación debe comprobar que el nuevo usuario introducido no existe ya en la base de datos, en caso contrario mostrará el mensaje de error al usuario.

REQ1.3	
Nombre	Almacenar usuario y contraseña
Descripción	Si la verificación del usuario y contraseña ha sido correcta, la aplicación almacenara la información en la base de datos.

US2 – Acceder a la aplicación

REQ2.1	
Nombre	Validar usuario y contraseña
Descripción	Cuando un usuario intente entrar en la aplicación, ésta validará con la base de datos que la información introducida se corresponde con alguna ya existente en el sistema para permitir el acceso.

REQ2.2	
Nombre	Mostrar Avatar
Descripción	Con el usuario y contraseña validados, la aplicación debe mostrar en pantalla el avatar almacenado en base de datos, mostrando, en el caso de nuevos usuarios un avatar definido por defecto, pero que no se encontrará almacenado en base de datos.

US3 – Crear/Modificar Avatar

REQ3.1	
Nombre	Modificar cara
Descripción	Una vez dentro de la aplicación, debe existir un botón llamado “Cara” cuya función es mostrar al usuario los diferentes tipos de cara y tonos de piel que puede tener el avatar. Pulsando sobre cada una de ellas, el avatar irá cambiando automáticamente según la selección del usuario.

REQ3.2	
Nombre	Modificar Pelo
Descripción	Una vez dentro de la aplicación, debe existir un botón llamado “Pelo” cuya función es mostrar al usuario los diferentes cortes de pelo y color del mismo que puede tener el avatar. Pulsando sobre cada una de ellas, el avatar irá cambiando automáticamente según la selección del usuario.

REQ3.3	
Nombre	Modificar Ojos
Descripción	Una vez dentro de la aplicación, debe existir un botón llamado “Ojos” cuya función es mostrar al usuario los diferentes tipos de ojos y colores de los mismos que puede tener el avatar. Pulsando sobre cada una de ellas, el avatar irá cambiando automáticamente según la selección del usuario.

REQ3.4	
Nombre	Modificar Nariz
Descripción	Una vez dentro de la aplicación, debe existir un botón llamado “Nariz” cuya función es mostrar al usuario los diferentes tipos de nariz que puede tener el avatar. Pulsando sobre cada una de ellas, el avatar irá cambiando automáticamente según la selección del usuario

REQ3.5	
Nombre	Modificar Boca
Descripción	Una vez dentro de la aplicación, debe existir un botón llamado “Boca” cuya función es mostrar al usuario los diferentes tipos de boca que puede tener el avatar. Pulsando sobre cada una de ellas, el avatar irá cambiando automáticamente según la selección del usuario

REQ3.6	
Nombre	Modificar Orejas
Descripción	Una vez dentro de la aplicación, debe existir un botón llamado "Orejas" cuya función es mostrar al usuario los diferentes tipos de orejas que puede tener el avatar. Pulsando sobre cada una de ellas, el avatar irá cambiando automáticamente según la selección del usuario

REQ3.7	
Nombre	Modificar Camiseta
Descripción	Una vez dentro de la aplicación, debe existir un botón llamado "Camiseta" cuya función es mostrar al usuario los diferentes tipos de camiseta y los colores de las mismas que puede tener el avatar. Pulsando sobre cada una de ellas, el avatar irá cambiando automáticamente según la selección del usuario

REQ3.8	
Nombre	Modificar Pantalón
Descripción	Una vez dentro de la aplicación, debe existir un botón llamado "Pantalón" cuya función es mostrar al usuario los diferentes tipos de pantalón y los colores de los mismos que puede tener el avatar. Pulsando sobre cada una de ellas, el avatar irá cambiando automáticamente según la selección del usuario

REQ3.9	
Nombre	Guardar Avatar
Descripción	La aplicación debe dar la posibilidad al usuario de guardar el avatar en cualquier momento del proceso de modificación, para evitar que se pierdan los cambios.

US4 – Mostrar Avatar en Juego

REQ4.1	
Nombre	Acceder a motor de juegos
Descripción	Se debe dotar a la interfaz del motor de OpenSpace, de la capacidad de conectarse a la base de datos de la aplicación a través del nombre de usuario.

REQ4.2	
Nombre	Mostrar Avatar en la pantalla seleccionada
Descripción	Con el nombre de usuario introducido, en caso de existir, se mostrará en el mapa de OpenSpace el avatar que exista almacenado en la base de datos.

REQ4.3	
Nombre	Simular movimiento del Avatar
Descripción	A partir de las características del avatar almacenado en la base de datos, éste podrá desplazarse correctamente por el mapa seleccionado.

4.4 Casos de uso

A continuación se van a especificar los casos de uso (CU a partir de ahora), que se encuentran basados en las US descritas anteriormente.

CU1	
Nombre	Alta de usuario nuevo.
Descripción	Se dará de alta un usuario no existente en la base de datos
Precondiciones	-
Pasos	1- Se pulsa la opción “nuevo usuario”.
	2- Se introduce un nombre de usuario y se pulsa “Validar”.
	3- Se introduce la contraseña dos veces y se pulsa “Registrarse”.
Resultado	La aplicación se abre y aparece un avatar definido por defecto.

CU2	
Nombre	Acceso de usuario existente.
Descripción	Se accederá a la aplicación a través de un usuario y existente.
Precondiciones	Usuario ya existente.
Pasos	1- Se pulsa la opción “Inicio Sesión”.
	2- Se introduce el nombre de usuario y contraseña y se pulsa “Entrar”.
Resultado	La aplicación se abre y aparece el avatar que hay guardado en la base de datos.

CU3	
Nombre	Modificación de Avatar.
Descripción	Se modifican las características del avatar y se guardan en la base de datos.
Precondiciones	Se debe haber accedido a la aplicación mediante usuario y contraseña.
Pasos	1- Se pulsa sobre la característica del avatar que se quiera modificar.
	2- Se cambia la opción de la característica o su color (en caso de que la característica tenga color) existente por la nueva deseada (el cambio aparece reflejado en el avatar)
	3- Se repiten los pasos 1 y 2 para cada característica que se desee modificar.
	4- Se pulsa "Guardar"
Resultado	El avatar almacenado en la base de datos ha sido modificado por el nuevo avatar que se muestra en la aplicación.

CU4	
Nombre	Navegación de Avatar en mapa.
Descripción	Se accederá a la interfaz del motor OpenSpace mediante el nombre de usuario y se seleccionara un mapa en el que se mostrara el avatar y se podrá navegar con él.
Precondiciones	Se debe haber creado un avatar.
Pasos	1- Introducir el usuario en la interfaz de OpenSpace.
	2- Seleccionar un mapa.
	3- Navegar por el mapa con el Avatar.
Resultado	El Avatar se mueve por el mapa sin problemas.

5. Diseño

Para cumplir con las necesidades que hemos detectado en el análisis, deberemos de disponer de una aplicación que sea capaz de permitir a un usuario lo siguiente:

- crear un avatar desde cero.
- modificar ese avatar si así lo desea.
- guardar la configuración del avatar para su posterior utilización.
- integrar el avatar en el motor de openspace.

Por lo tanto, no solo se tendrá que crear una aplicación capaz de configurar un avatar al gusto del usuario, sino que además, se tendrán que desarrollar las herramientas necesarias para la correcta integración del avatar en el juego.

5.1 El avatar

La aplicación que permite la creación y modificación del avatar, es un módulo desarrollado en Adobe Flash CS5 [19], que se basa en una navegación por botones, que permite al usuario logarse para acceder a su propio perfil de avatar. Si no se ha registrado anteriormente, la aplicación solicita el ingreso a través de un usuario no repetido y una contraseña. A partir de que concede el acceso a la creación del avatar, presenta al usuario distintas opciones de caracterización del avatar y la posibilidad de guardar los cambios realizados o salir de la aplicación en cualquier momento.

Ese es el funcionamiento básico de la aplicación, y a continuación se va a explicar la navegación de pantallas, así como la estructura de la pantalla y las diversas zonas que la componen

5.1.1. Navegación de pantallas

Nos encontraremos primero con una pantalla de inicio que nos pedirá registrarnos o iniciar sesión en caso de que ya nos hayamos registrado con anterioridad. Se pueden registrar tantos usuarios como se quieran, con la única restricción de que el

usuario a registrar no haya sido ya creado, no diferenciando el nombre del usuario entre mayúsculas y minúsculas. La contraseña sí diferencia entre mayúsculas y minúsculas.

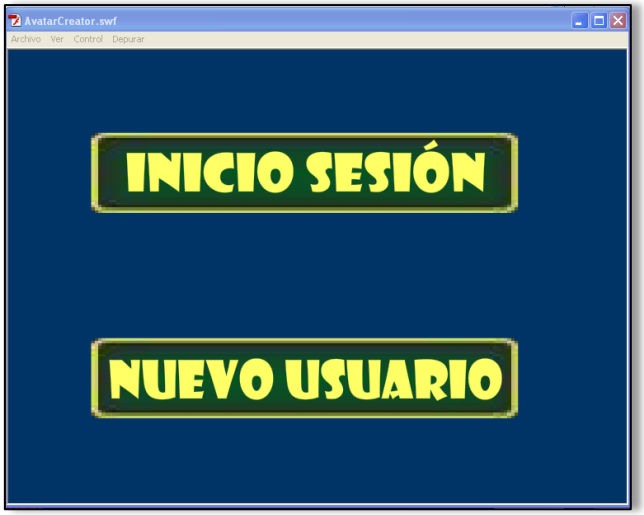


Ilustración 29. Pantalla de Inicio

Si elegimos crear un nuevo usuario, tendremos acceso a una pantalla que validará el nombre de usuario, y si este es válido, pasaremos a otra pantalla que nos obligará a crear una contraseña.

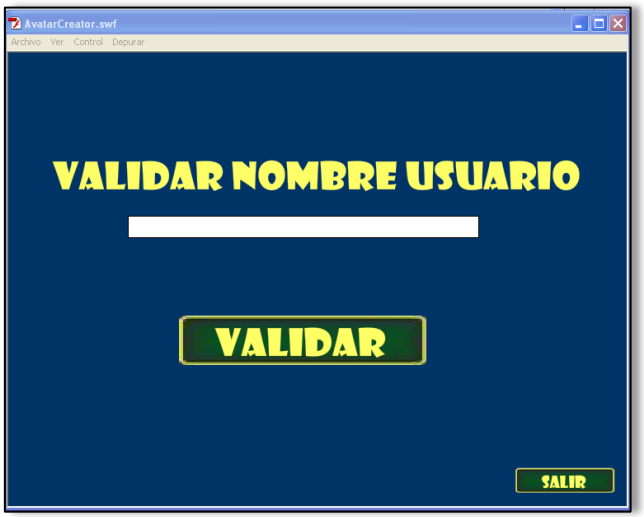


Ilustración 30. Pantalla de Validación de nuevo usuario

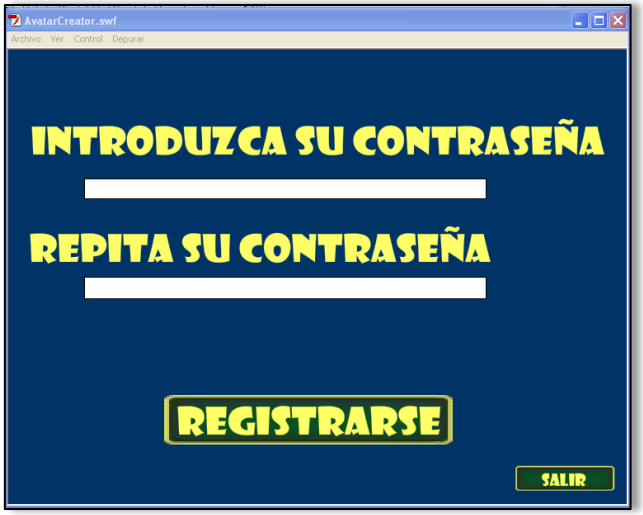


Ilustración 31. Pantalla de Creación de contraseña

Si ya tenemos un usuario creado y hemos decidido iniciar sesión, la pantalla que se nos abrirá, será la típica pantalla de acceso con contraseña.

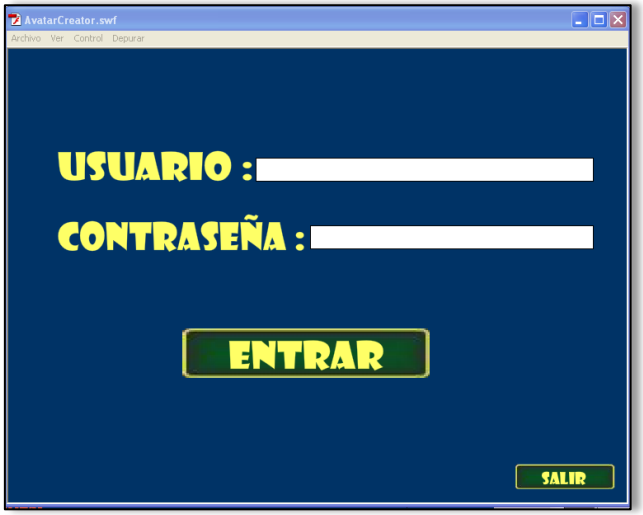


Ilustración 32. Pantalla de Login

Una vez se ha comprobado la identidad el usuario y se ha iniciado sesión, se presenta al usuario la pantalla principal de la aplicación con el avatar por defecto si es la primera vez que se entra, o el avatar definido por el usuario en su última ejecución.

Llegados a esta pantalla, aparecerán cuatro zonas diferentes en cuanto a la funcionalidad para el usuario:

- Una zona fija de botones a la izquierda de la pantalla en la que el usuario podrá elegir la característica que desea modificar.

Ilustración 33. Botones de Características

- Una zona central fija también, en la que va a aparecer el avatar con las características que se le están asignando, y que irá actualizándose según se le vayan asignando diferentes tipos de características y colores.

Ilustración 34. Espacio del Avatar

- Una zona variable de opciones a la derecha del avatar, que irán cambiando según la característica que se seleccione para modificar. Esa zona variable incluye los diferentes tipos dentro de cada característica, así como una selección de colores para las características que puedan ser coloreadas.

Ilustración 35. Opciones de cada característica

- Una zona inferior, en parte fija y en parte variable, que contendrá dos botones (uno en cada esquina de la pantalla) para guardar los cambios realizados y para salir de la sesión. También incluye una zona en la que se

podrá ver el nombre del usuario que se encuentra en la sesión y en la que podrán aparecer distintos mensajes de aviso.

Ilustración 36. Guardado, salida y mensajes al usuario

5.2 La base de datos

La base de datos nos debe permitir guardar las diferentes características del avatar, así como la información correspondiente al usuario con el que se corresponda dicho avatar.

También ha de servirnos para guardar las diferentes puntuaciones que el usuario consiga en cada una de sus conexiones, y darnos la posibilidad de que un mismo usuario pueda participar en varios juegos.

Como parte de la integración de la aplicación, la base de datos también nos debe permitir almacenar todos los juegos que se vayan a instalar y sobre los que el usuario podrá jugar.

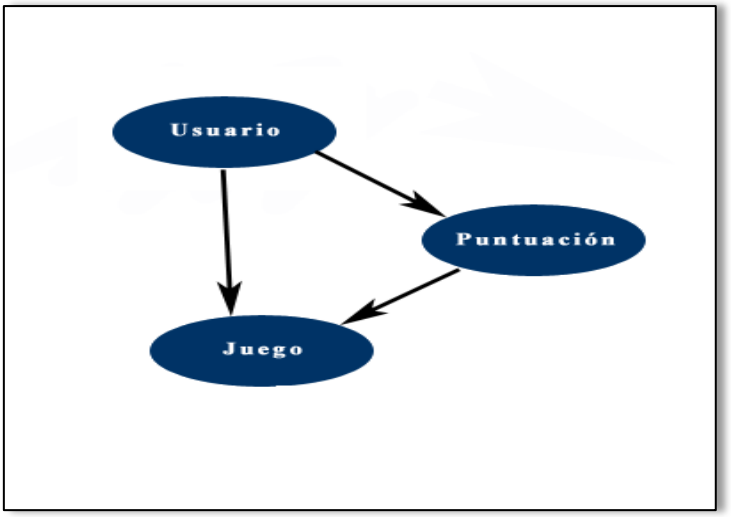


Ilustración 37. Modelo de entidades de la BBDD

La base de datos constará de 4 tablas en las que guardaremos toda la información necesaria para almacenar el avatar creado por el usuario, así como sus datos de acceso y las diferentes puntuaciones obtenidas en los distintos juegos en los que haya participado, como se puede ver en la siguiente imagen.

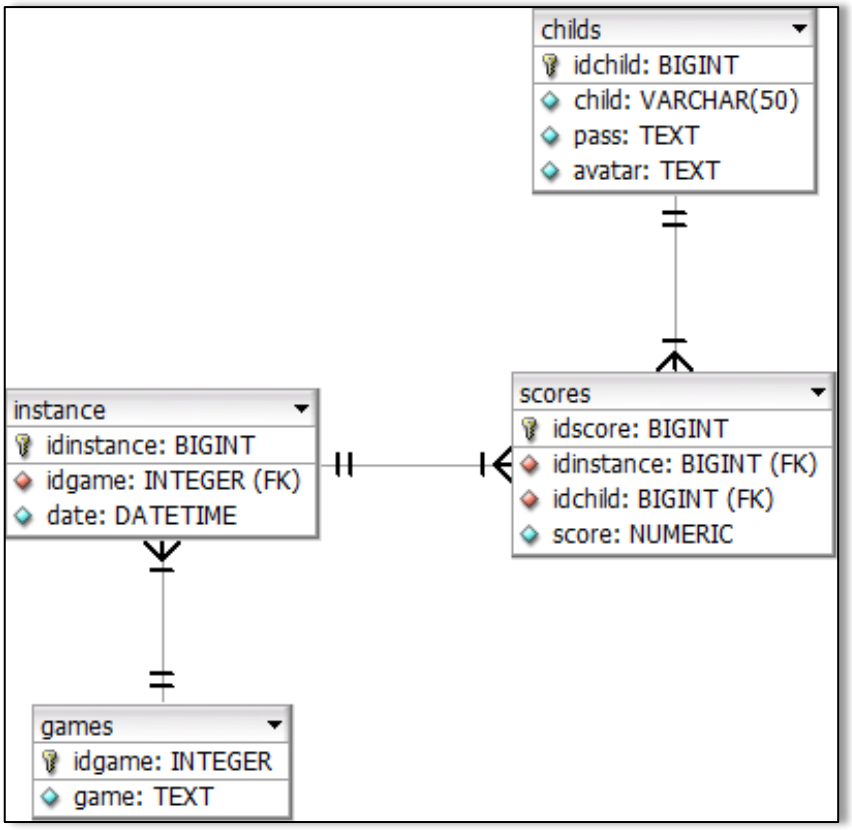


Ilustración 38. Esquema de la base de datos

TABLA CHILDS

La tabla **childs** contendrá la información básica que corresponde al usuario y que le permitirá acceder a los diferentes juegos, así como la creación y modificación de su propio avatar.

- **idchild**: clave primaria de la tabla.
- **child**: nombre de usuario. Es el nombre con el que se ha registrado y por el que será identificado tanto en la aplicación de personalización de avatares como en los diferentes juegos a los que acceda. El nombre de usuario es único y no puede estar repetido, no diferenciando entre letras mayúsculas y minúsculas.
- **pass**: contraseña de acceso del usuario. Es la contraseña que en combinación con el nombre de usuario, dará al usuario acceso a la modificación de su avatar y a la participación en los juegos. La contraseña si distingue entre mayúsculas y minúsculas.

- **avatar**: datos del avatar creado por el usuario. Contiene las características del avatar creado por el usuario, especificando en cada una de ellas el código de la característica y su color, en el caso de que esta lo requiera. A pesar de ser un campo de texto, está formado por una estructura de xml para un tratamiento más eficiente en la aplicación.

El formato XML es el siguiente:

Ilustración 39. Formato xml del avatar

TABLA SCORES

La tabla **scores** contendrá la información acerca de las puntuaciones realizadas por el usuario en cada juego cada vez que haya accedido a él, con lo que se podrán obtener en el juego un histórico de puntuaciones para ver la evolución del usuario en el juego.

- **idscore**: clave primaria de la tabla.

- **idinstance**: identificador de la instancia de juego en la que el usuario ha realizado la puntuación almacenada. Es una clave foránea proveniente de la tabla **instance**.
- **idchild**: identificador del usuario que ha realizado la puntuación almacenada. Es una clave foránea proveniente de la tabla **childs**.
- **score**: puntuación realizada por el usuario en el juego.

TABLA INSTANCE

La tabla **instance** servirá para relacionar las puntuaciones de un usuario con el juego al que correspondo y el momento en el que tuvo lugar el acceso.

- **idinstance**: clave primaria de la tabla.
- **idgame**: identificador del juego en el que el usuario ha jugado. Es una clave foránea proveniente de la tabla **games**.
- **date**: fecha de juego del usuario.

TABLA GAMES

La tabla **games** contendrá una lista de todos los juegos que estén o hayan estado disponibles para los usuarios.

- **idgame**: clave primaria de la tabla.
- **game**: nombre del juego.

5.3 La integración del avatar en los juegos

El avatar que genera la aplicación deberá poderse integrar correctamente en el entorno del juego, que en este caso será el entorno de OpenSpace.

Para ello, se deben crear todas las secuencias de imágenes de cada una de las características que se hayan definido en la aplicación. Para crear dichas secuencias de

imágenes se han de crear primero las diferentes imágenes estáticas correspondientes a cada característica, teniendo en cuenta que las posibles vistas del avatar son Norte, Noreste, Este, Sureste, Sur, Suroeste, Oeste y Noroeste, y que dentro de cada vista el avatar puede realizar el movimiento de caminar, que dependiendo de la suavidad que le queramos dar al movimiento, puede oscilar entre 3 (movimiento muy brusco) y 7 (movimiento muy suave).

En nuestro caso, nos hemos ido a un movimiento normal con 5 posiciones fijas que formaran una secuencia y simularan el movimiento al andar del avatar.

- **NORTE**

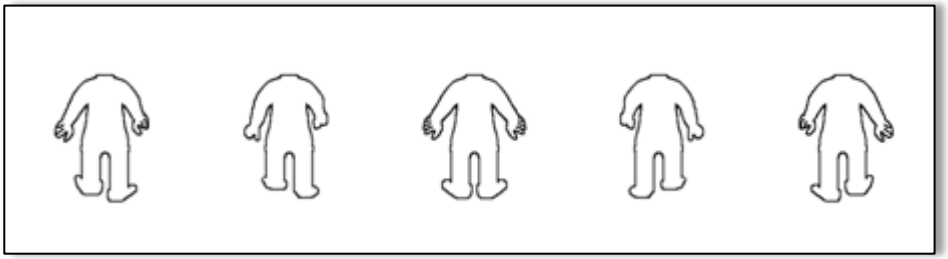


Ilustración 40. Movimiento de Avatar en dirección Norte

- **NORESTE**

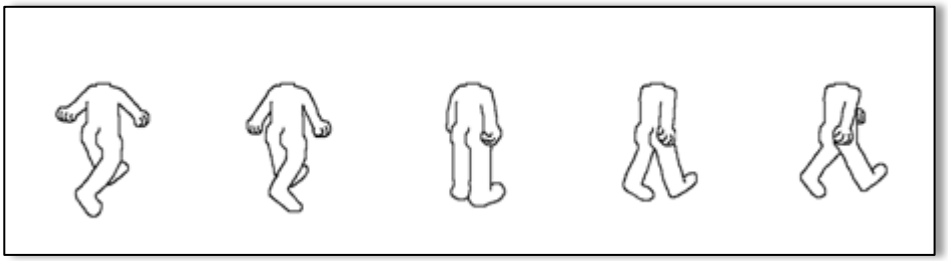


Ilustración 41. Movimiento de Avatar en dirección Noreste

- **ESTE**

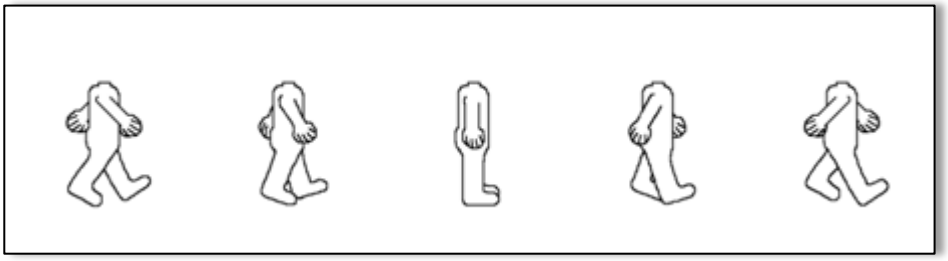


Ilustración 42.Movimiento de Avatar en dirección Este

- **SURESTE**

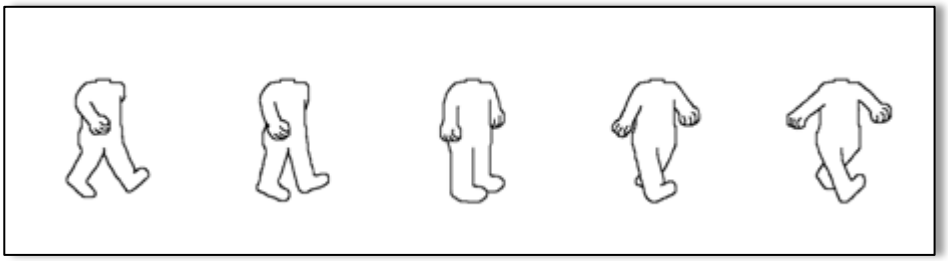


Ilustración 43. Movimiento de Avatar en dirección Sureste

- **SUR**

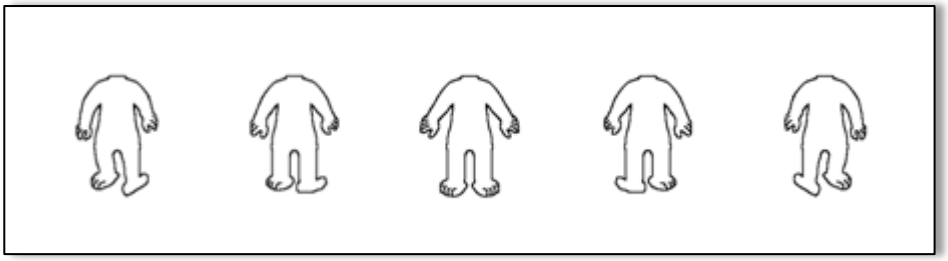


Ilustración 44. Movimiento de Avatar en dirección Sur

- **SUROESTE**

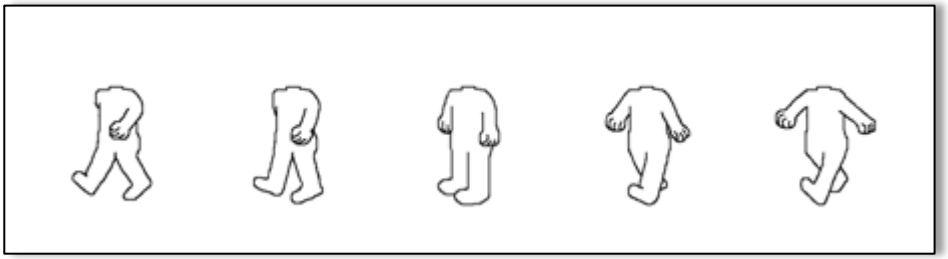


Ilustración 45. Movimiento de Avatar en dirección Suroeste

- **OESTE**

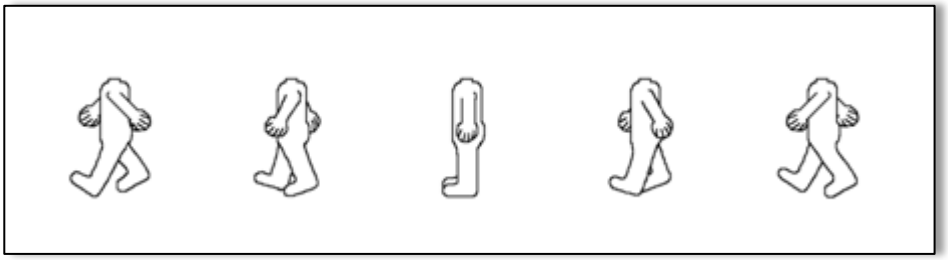


Ilustración 46. Movimiento de Avatar en dirección Oeste

- **NOROESTE**

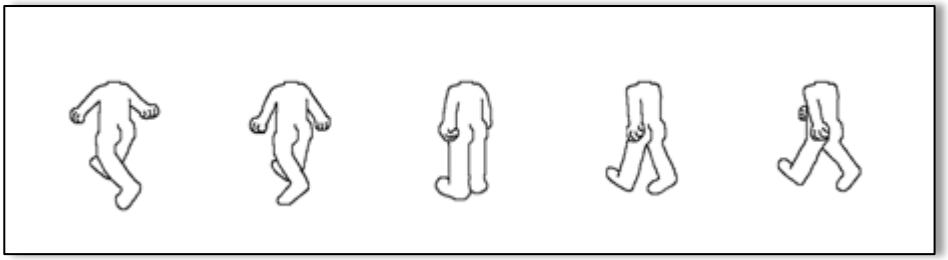


Ilustración 47. Movimiento de Avatar en dirección Noroeste

5.4 Diseño de los dibujos

Como la aplicación está destinada a un público infantil, se ha optado por realizar unos diseños de avatar poco complejos y de aspecto aniñado, con el fin de una mayor identificación del usuario final con el avatar.

Tampoco se ha querido dotar de demasiada ornamentación al avatar, para no crear un exceso de posibilidades que hagan la aplicación menos manejable de cara al usuario final.

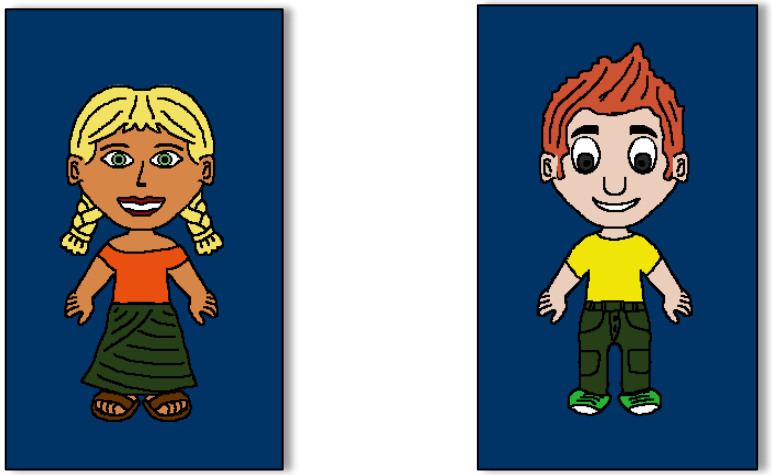


Ilustración 48. Modelos de los Avatares

6. Implementación y configuración

6.1 Estructura de la aplicación

La estructura de la aplicación está realizada en función de las posibilidades que nos otorga Flash, tratando las distintas pantallas de la aplicación como si de fotogramas de una película se tratase, y montando en cada fotograma una sucesión de capas que dota a la aplicación de las funcionalidades requeridas.

A continuación se muestran todos los fotogramas de los que consta la aplicación:

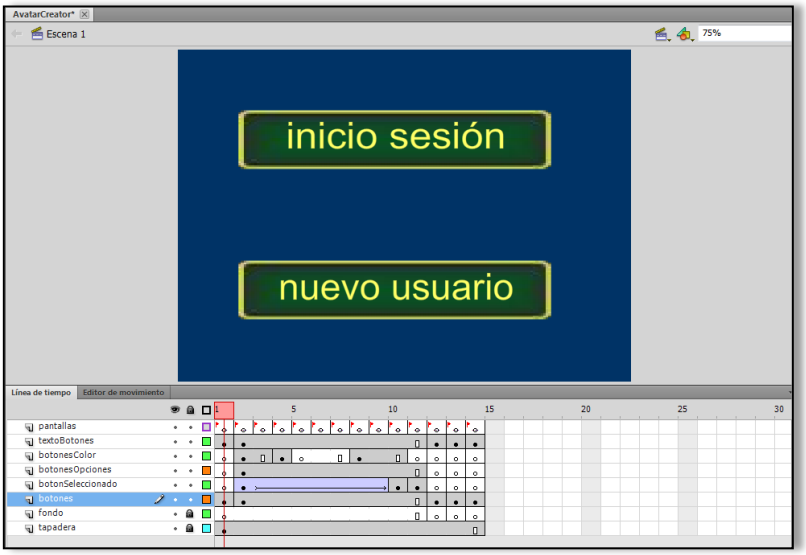


Ilustración 49. Fotograma 1

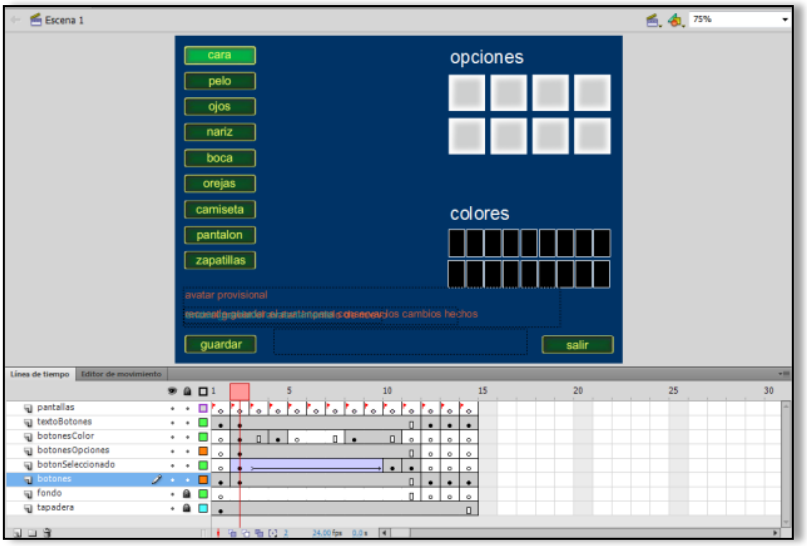


Ilustración 50. Fotograma 2

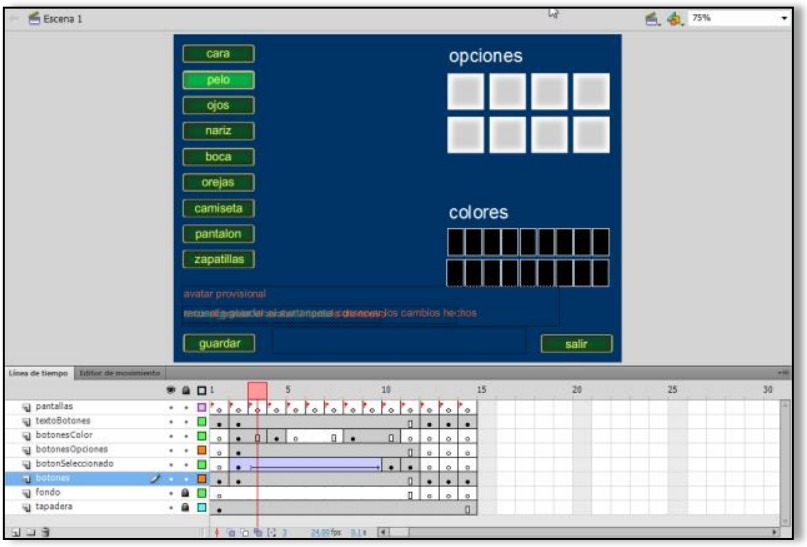


Ilustración 51. Fotograma 3

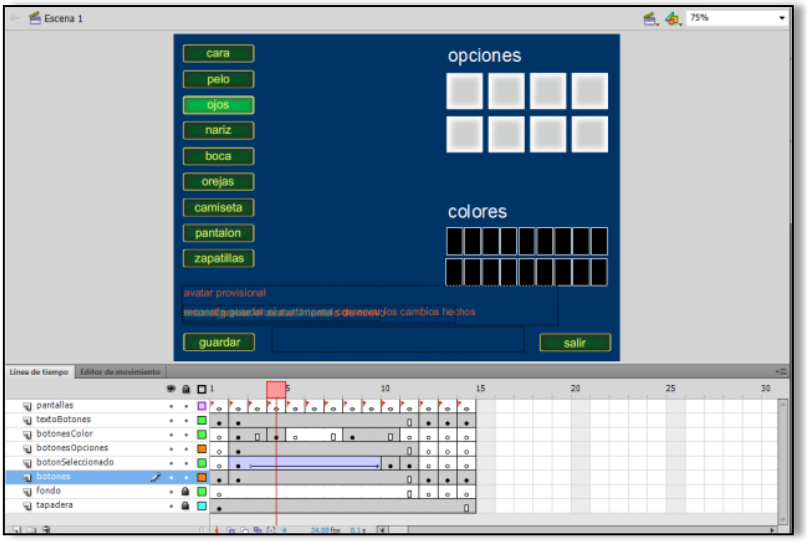


Ilustración 52. Fotograma 4

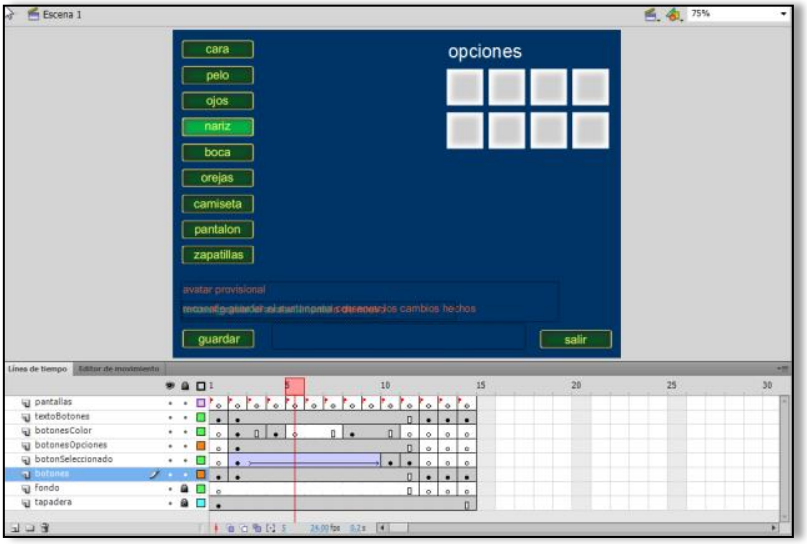


Ilustración 53. Fotograma 5

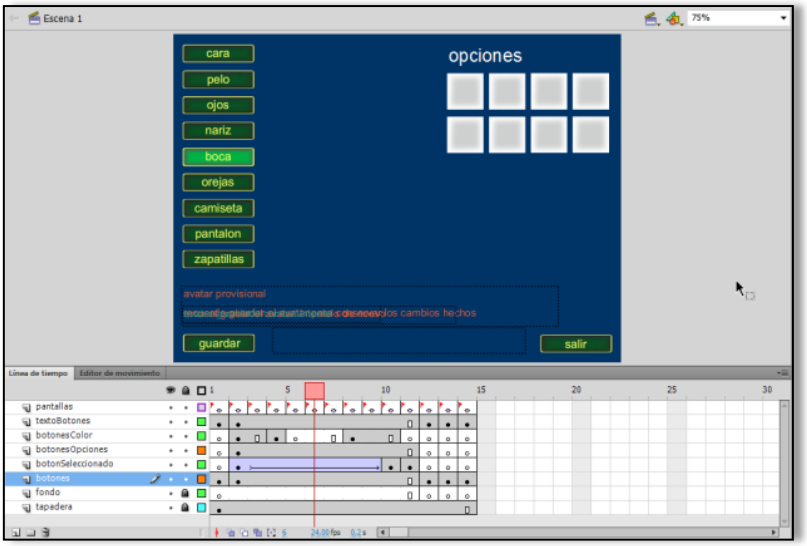


Ilustración 54. Fotograma 6

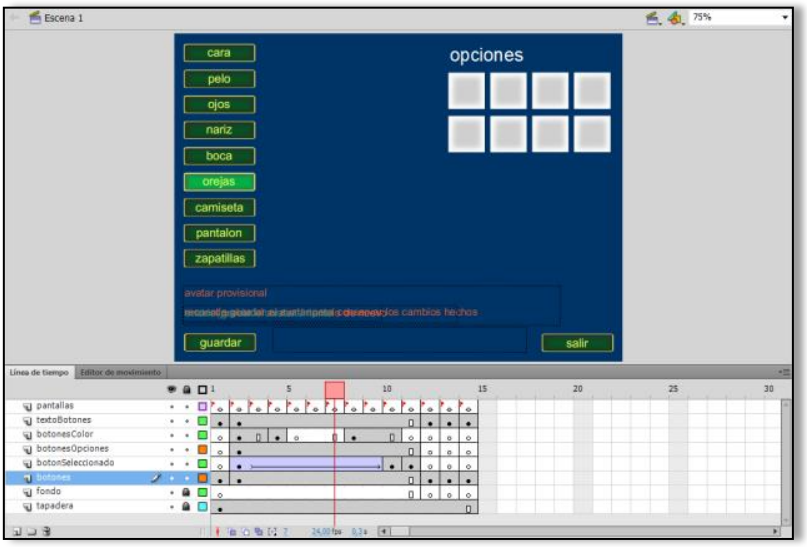


Ilustración 55. Fotograma 7

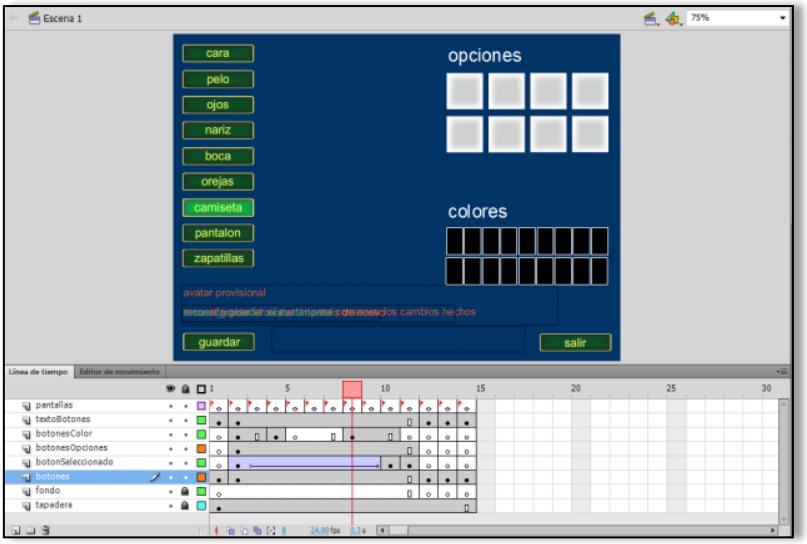


Ilustración 56. Fotograma 8

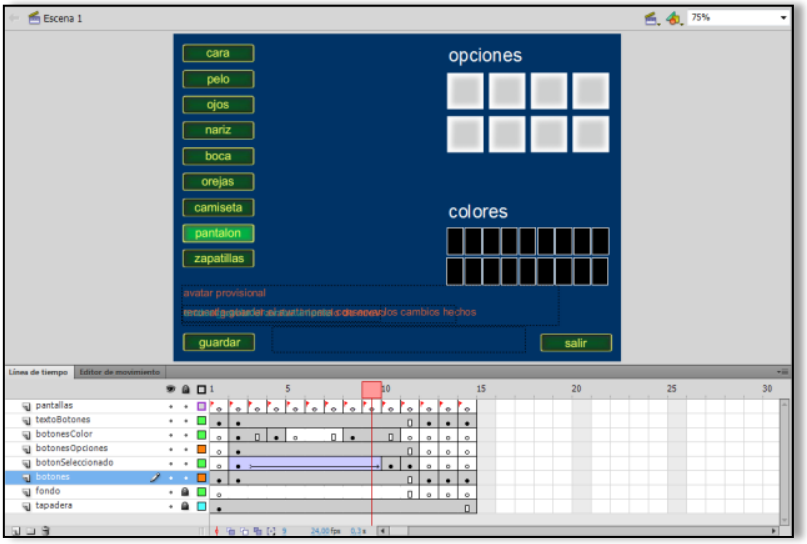


Ilustración 57. Fotograma 9

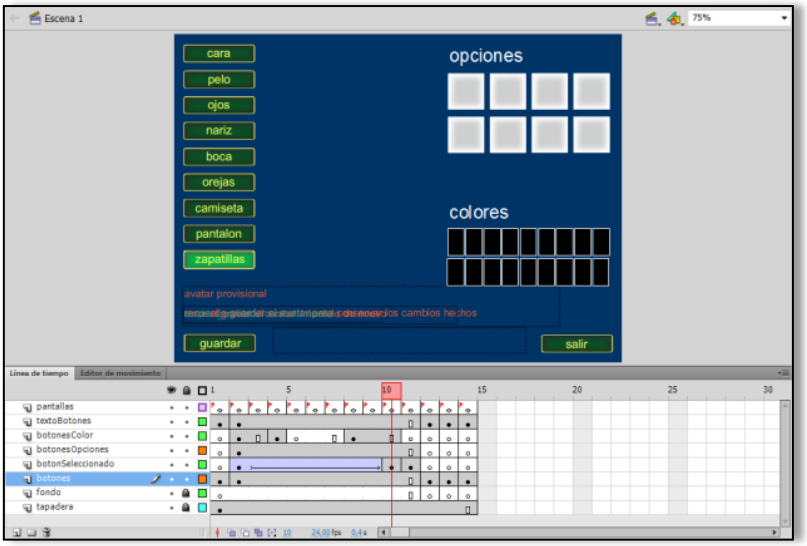


Ilustración 58. Fotograma 10

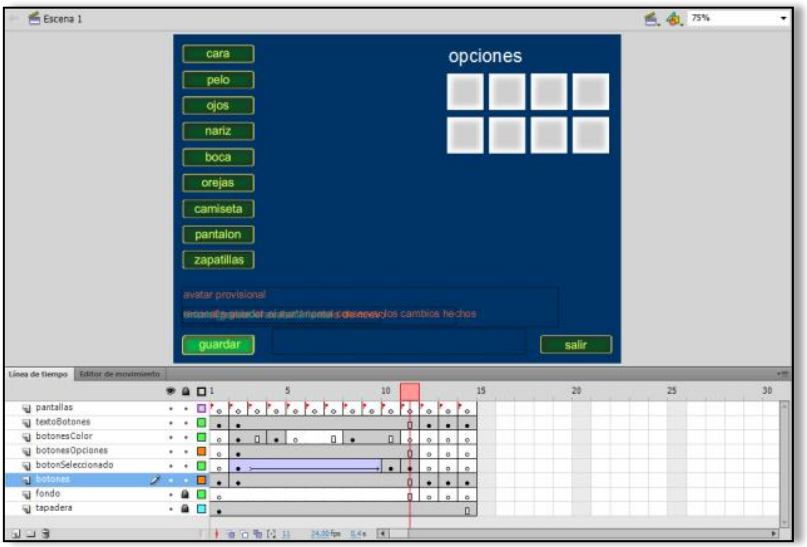


Ilustración 59. Fotograma 11

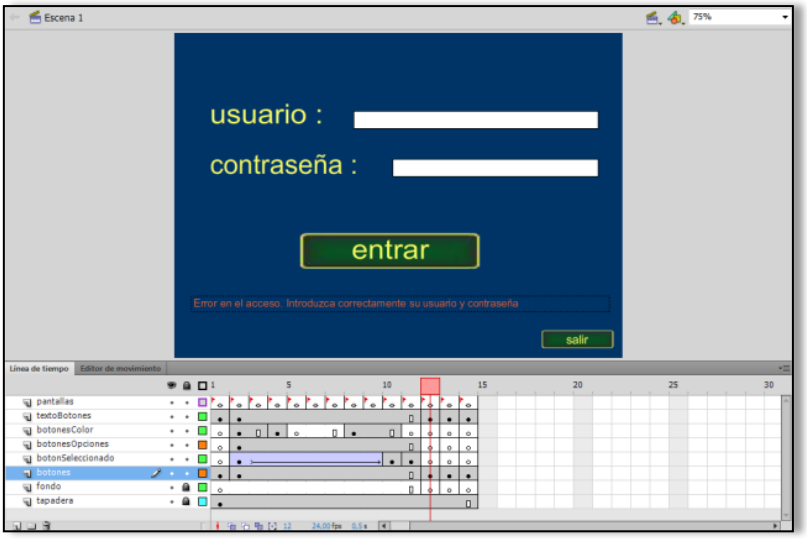


Ilustración 60. Fotograma 12

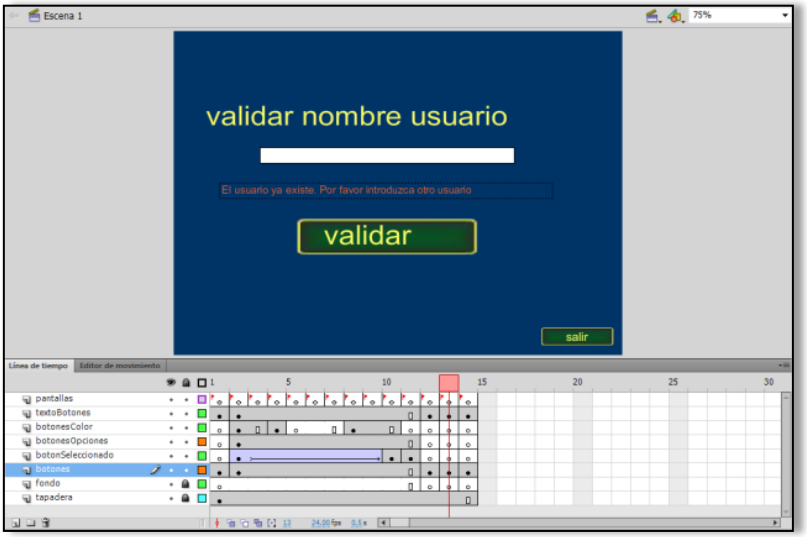


Ilustración 61. Fotograma 13

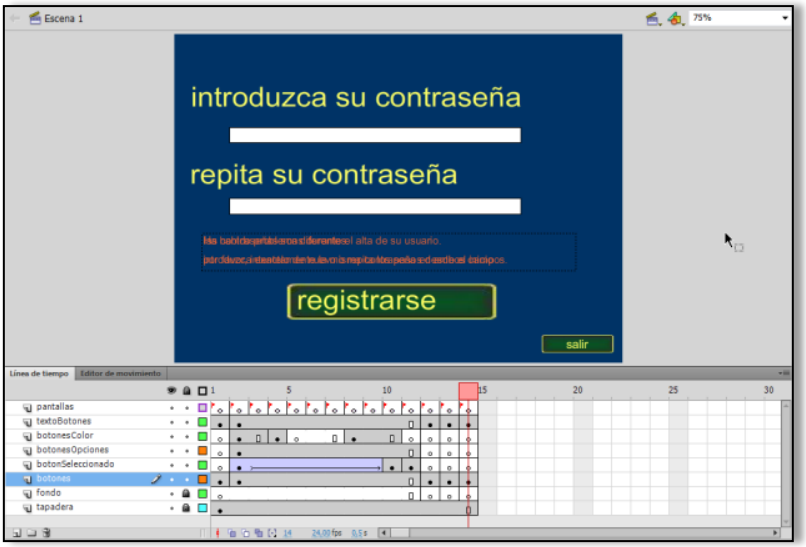


Ilustración 62. Fotograma 14

Como se puede observar en las imágenes, para cada fotograma intervienen las distintas capas de las que se compone dicho fotograma.

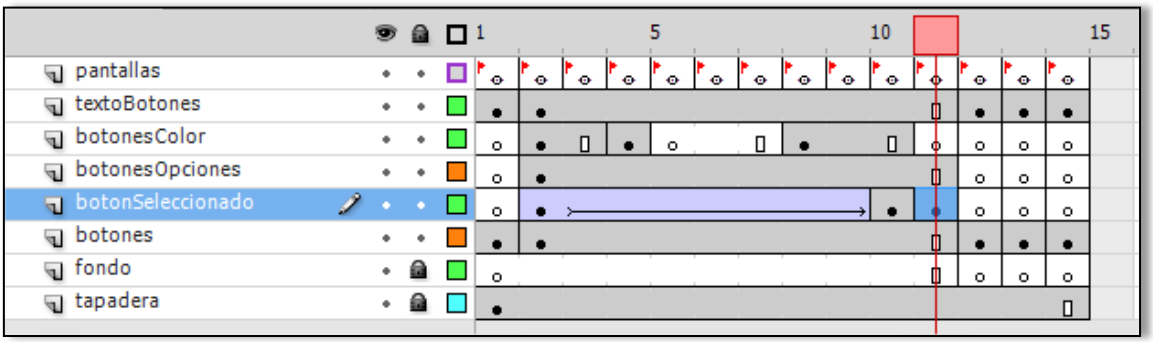


Ilustración 63. Matriz de capas y fotogramas

El funcionamiento de cada capa es el siguiente:

pantallas: se utiliza para identificar cada fotograma, de manera que cada vez que nos queramos situar en un fotograma en concreto lo haremos según se encuentra identificado en dicha capa.

textobotones: se utiliza para que se visualice en cada botón el texto correspondiente al fotograma.

botonesColor: se utiliza para controlar los botones que modifican el color de cada una de las características que son susceptibles de un cambio de color.

botonesOpciones: se utiliza para controlar las opciones de elección posibles en cada característica del avatar, de manera que dependiendo del fotograma en el que se encuentre se muestren las opciones de elección correctas.

botonSeleccionado: se utiliza para resaltar la característica del avatar en la que nos encontramos.

botones: se utiliza para mostrar en cada fotograma los botones necesarios para el funcionamiento de la aplicación.

fondo: se utiliza para mantener el fondo constante en todos los fotogramas.

El lenguaje de ActionScript 3, nos permite dotar a los objetos (botones, opciones) de una funcionalidad general mediante la vinculación de una clase de código AS3, que dependiendo del fotograma en el que nos encontremos (para ello utilizamos la propiedad de la capa pantallas) nos permitirá utilizar las diversas funcionalidades desarrolladas en clases externas no vinculadas con objetos de los fotogramas y en los cuales se ha desarrollado el conjunto de la funcionalidad de la aplicación.

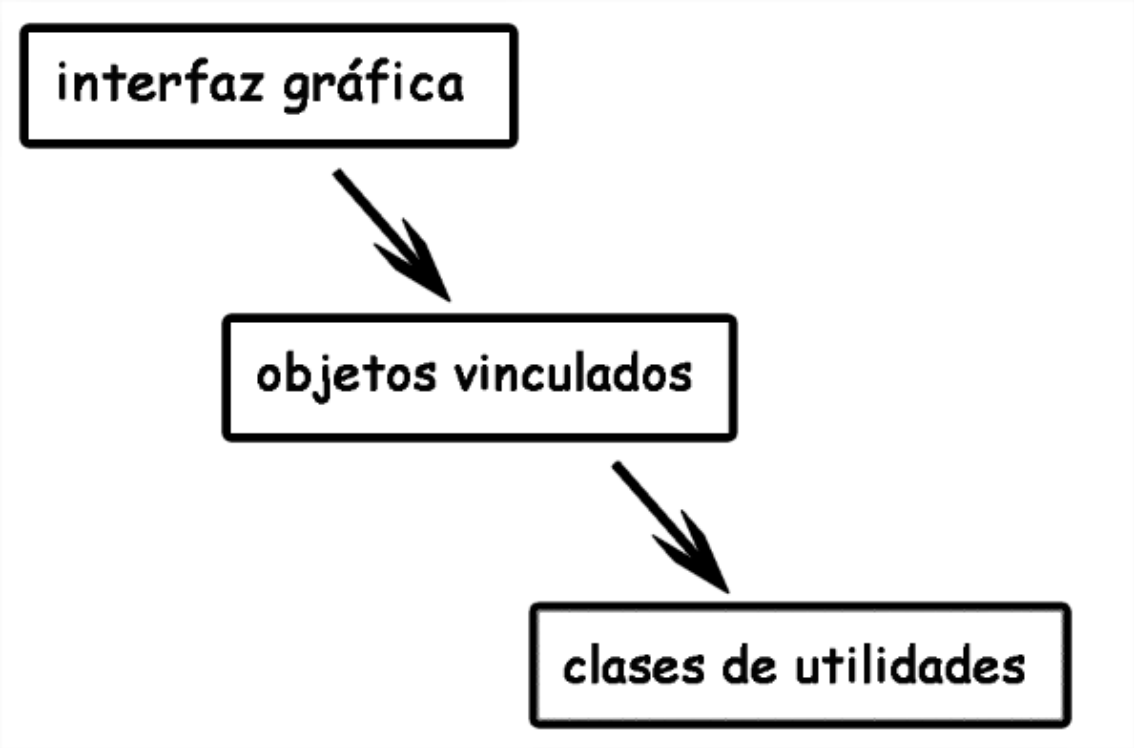


Ilustración 64. Esquema ActionScript

6.2 Dibujo de los avatares

Para el dibujo de los avatares se ha utilizado la herramienta Adobe Photoshop. Se ha aprovechado su sistema de capas para una vez dibujado el modelo base del avatar, ir añadiendo características al mismo en capas superpuestas.

Para ello, primero se definió el perfil del avatar que íbamos a utilizar, y como la aplicación va a destinada a un público infantil, se estudiaron los distintos tipos de dibujos animados que se presentan hoy en día para dicho público. Se tuvieron en cuenta desde los dibujos animados más actuales de estilo manga, anime, hasta los dibujos más clásicos al estilo de Zipi y Zape.

Tras un par de bocetos de cada distinto estilo, al final se decidió por crear un dibujo que tuviese una base clásica con un modelo de avatar de aspecto aniñado en vez de optar por un diseño más futurista y con más detalles como habría supuesto irse más hacia un estilo de dibujo oriental tipo manga o anime.

También se ha tenido en cuenta para la realización del dibujo, el hecho de que en la representación de los avatares en los juegos, la resolución de los dibujo tiene un tamaño que no permitiría la apreciación de según qué detalles de los dibujos.

Al tratarse de un dibujo dirigido al público infantil, se decidió el hecho de no hacer un cuerpo para chico y otro para chica, y buscar un modelo que sirviese para ambos sexos.

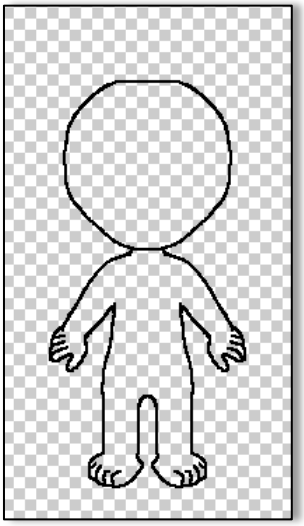


Ilustración 65. Modelo de cuerpo

Como hemos dicho antes, una vez definido el modelo de avatar a dibujar, y aprovechando el sistema de capas que nos facilita Adobe Photoshop, se procedió a crear todas y cada una de las diferentes características del avatar. Para ello, y con la capa base por debajo, dibujábamos encima del modelo, la característica que nos ocupase en ese momento teniéndola ya colocado y evitando tener que hacer ajustes de posición y proporción una vez hecho el dibujo.

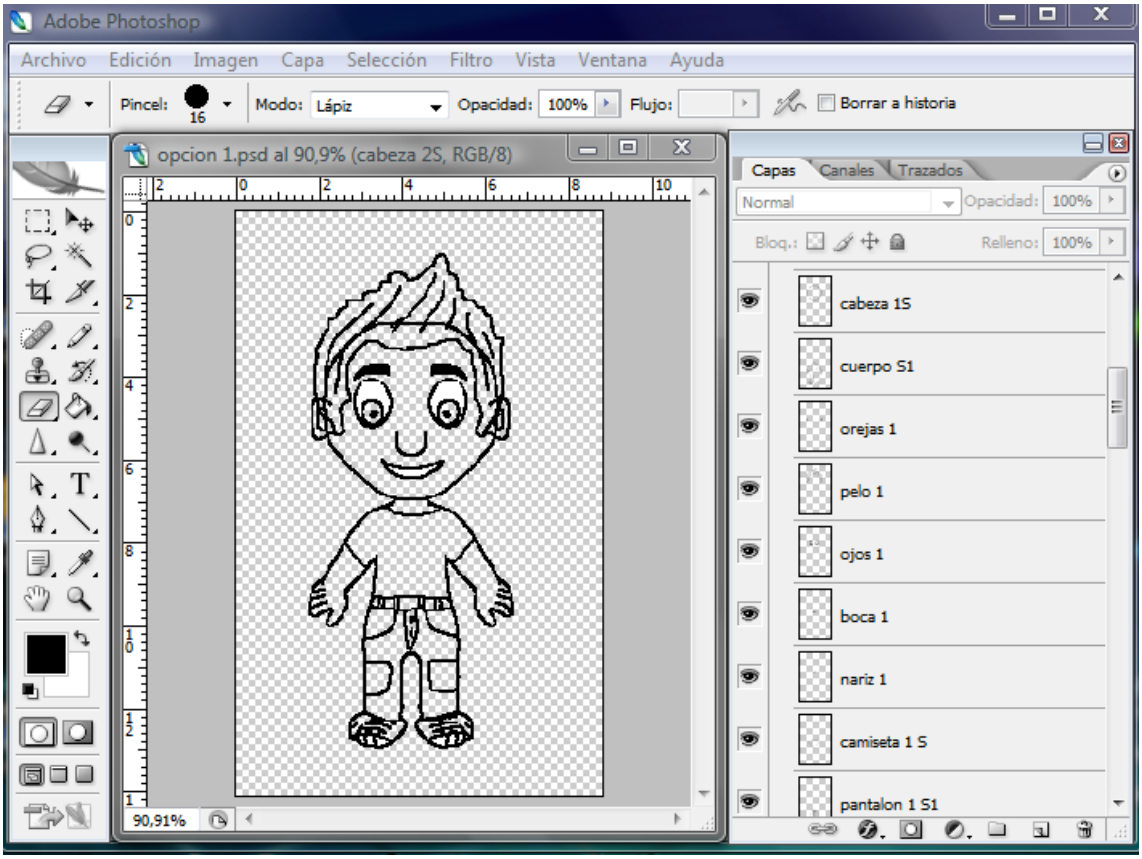


Ilustración 66. Modelado de Avatar con Photoshop

Para poder configurar los colores de las distintas características, la solución pasa por hacer un dibujo igual al que se quiere colorear, y rellenarlo de color negro, para que, a la hora de asignarle un color desde la aplicación, éste no se vea alterado por el color de fondo de la característica. El motivo de que el color de relleno sea negro y no otro, se debe a que el negro es la ausencia de color, y así no se produce variación del color elegido.

Ilustración 67. Relleno de color de los objetos

Empezando desde arriba, las características y sus diferentes opciones son las siguientes:

- **PELO:**

Ilustración 68. Modelos de pelo

- **CARA:**

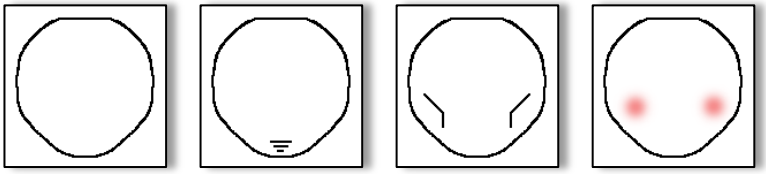


Ilustración 69. Modelos de cara

- **OJOS:**

Ilustración 70. Modelos de ojos

- **NARIZ:**

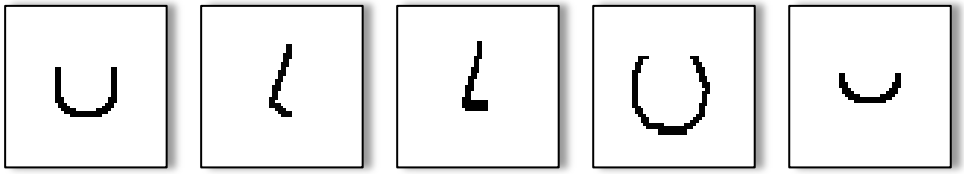


Ilustración 71. Modelos de nariz

- **OREJAS:**

Ilustración 72. Modelos de orejas

- **BOCA:**

Ilustración 73. Modelos de boca

- **CAMISETA:**

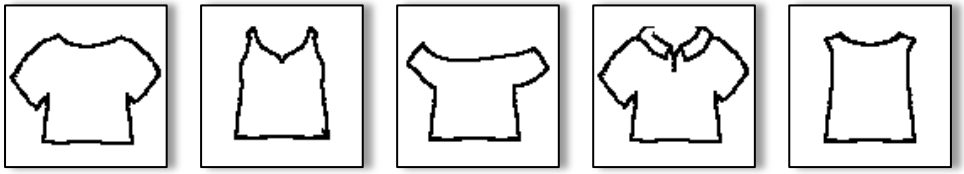


Ilustración 74. Modelos de camiseta

- **PANTALÓN:**

Ilustración 75. Modelos de pantalón

- **ZAPATILLAS:**

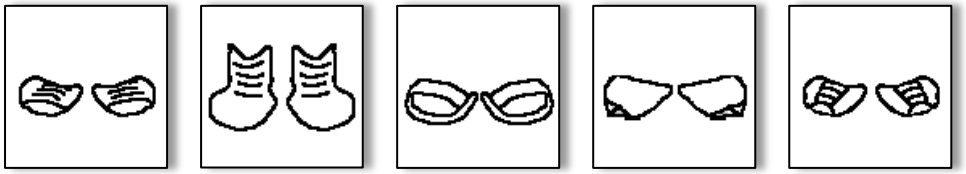


Ilustración 76. Modelos de zapatillas

Una vez realizados los diseños de todas las características del avatar, hay que dibujar los diferentes puntos de vista desde los que se verá el avatar, así como cada una de las posiciones al caminar del avatar y su efecto sobre las características.

Como se ha indicado antes, hemos elegido un movimiento de 5 posiciones, y cada posición deberá tener los 8 puntos de vista, por lo que tendrá que haber aproximadamente 80 dibujos por cada característica, ya que habrá que hacer el dibujo de cada característica y su correspondiente relleno de color. Algunas características no necesitan tener las 5 posiciones de movimiento ya que el movimiento no afecta a su forma, al igual que otras no necesitaran tener los 8 puntos de vista, ya que según de qué característica se trate, para el usuario la característica permanece en la parte oculta del dibujo.

6.3 Integración de los avatares

Una vez realizados todos los dibujos, hay que dotarlos de movimiento. Para ello, debemos crear películas de movimiento con la ayuda de Adobe Flash.

Estas películas estarán compuestas por la secuencia de los dibujos que previamente hemos realizado, y para evitar que el movimiento sea imperceptible duplicaremos cada posición.

La secuencia de imágenes guarda el siguiente patrón:

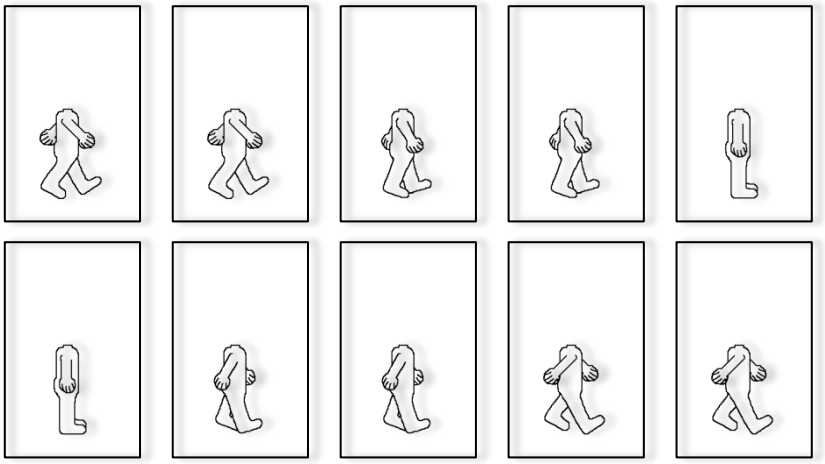


Ilustración 77. Secuencia de movimiento

Con estas premisas, hay que crear un archivo de flash (*.swf) en el que se encuentren todas las secuencias de imágenes de cada característica del avatar.

Por cada característica y opción, deberán existir al menos dos secuencias de imágenes, una que corresponda a la secuencia de imágenes del avatar en movimiento y otra que corresponda a las imágenes del avatar en posición estática. En el caso de que la característica lleve algún color asignado, el número de secuencias de imágenes por opción y característica seria de cuatro.

Cada secuencia de imágenes debe ir vinculada a una clase, para que la aplicación pueda hacer referencia a ellas. En este caso hemos optado por una nomenclatura que incluye el nombre de la característica, el número de opción dentro de la característica, la propiedad de color o su ausencia, y por último, el estado de movimiento en el que se encuentra el avatar, si está caminando (walk) o permanece quieto (stand).

característica_numeroopcion_[*color*_]estado

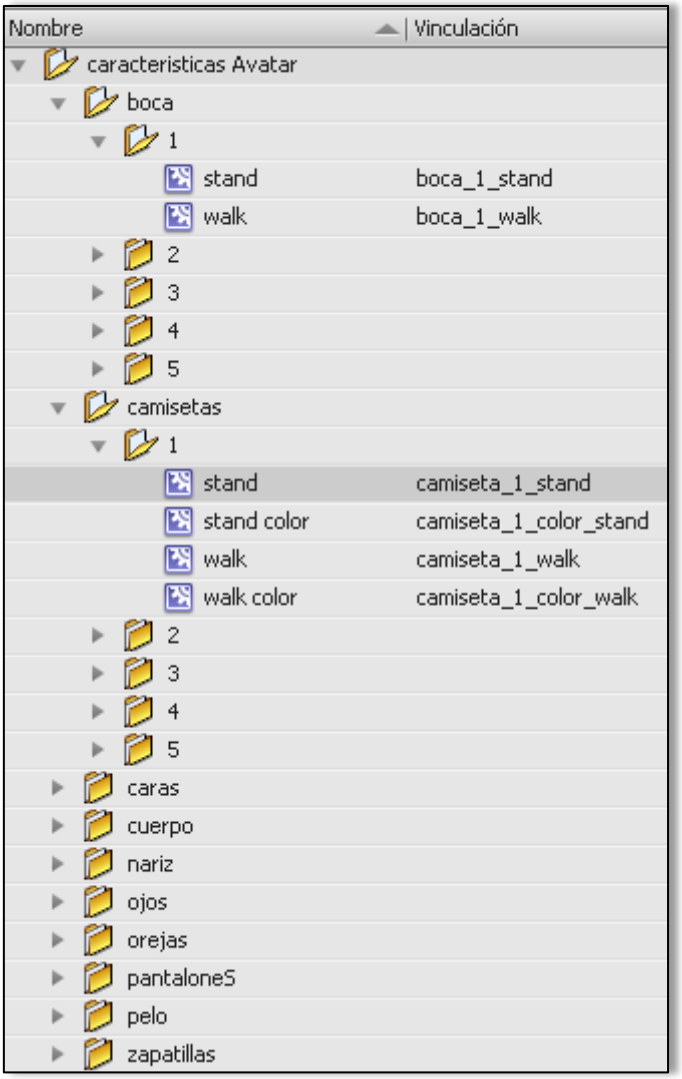


Ilustración 78. Clases de las características

La formación y número de fotogramas de cada secuencia de imágenes, dependerá del tipo de posición del que trate dicha secuencia:

- Para las secuencias de imágenes en las que el avatar se encuentra caminando, necesitaremos como hemos explicado antes, dos fotogramas por posición, (de las 5 posiciones que hemos dicho íbamos a hacer constar cada movimiento), y para simular un movimiento completo, para cada dirección (Norte, Sur,…) deberemos crear un ciclo completo que comience en la posición

inicial de posicionamiento estático y que termine justo en la posición anterior.

La secuencia quedaría de la siguiente manera:

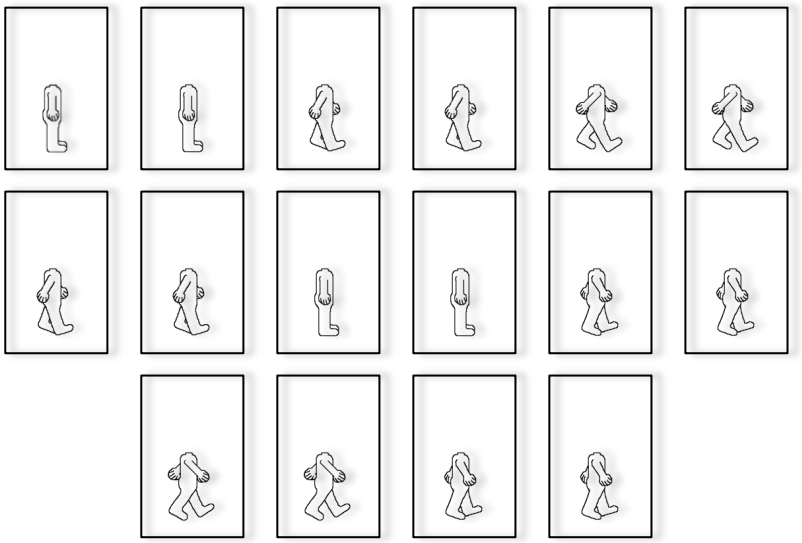


Ilustración 79. Secuencia de movimiento

Como se puede observar abajo, para cada dirección necesitamos un bucle de movimiento.

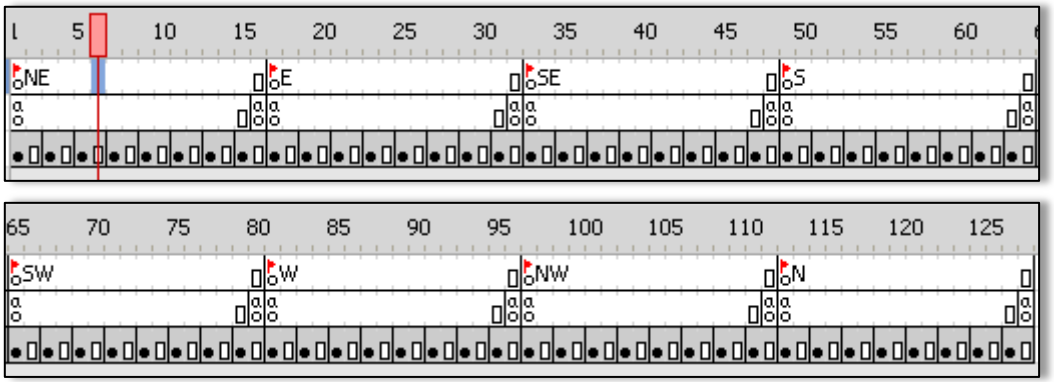


Ilustración 80. Bucles de movimiento por dirección

Por último, la aplicación flash que va a generar esas secuencias de movimiento, debe incluir una clase que extienda de avatar, y que será la que se encargue de gestionar las diferentes posiciones y movimientos del avatar, y de "pintarlas" en el juego (en nuestro caso avatarDrawer). Para ello, en dicha clase se ha de generar un método que se encargue de recoger los datos que contenga el avatar y que habrán sido recogidos en la conexión al juego, y dibujarlos uno a uno en la dirección y la posición correcta.

Para evitar que unos elementos tapen a otros a la hora de dibujarlos en el juego, el orden de dibujo de las características tiene que tener un criterio por el cual se dibujen primero las capas del fondo y se vaya dibujando cada vez una capa superior. Para dotar de color a cada característica, primero se le asigna el color y después se dibuja el contorno de la característica. El orden seguido para el dibujo de este avatar es el siguiente:

- 1 Color del cuerpo.
- 2 Cuerpo.
- 3 Color de las zapatillas.
- 4 Zapatillas.
- 5 Color de los pantalones.
- 6 Pantalones.
- 7 Color de la camiseta.
- 8 Camiseta.
- 9 Color de la cara.
- 10 Cara.
- 11 Boca.
- 12 Color de la nariz.
- 13 Nariz.
- 14 Color de los ojos
- 15 Ojos
- 16 Color del pelo.
- 17 Pelo.
- 18 Color de las orejas.
- 19 Orejas.

6.4 Configuración del entorno

El proyecto consiste en la aplicación realizada con Flash y el servidor de SmartFox por lo que hay que configurar ambos componentes de manera que el modulo funcione correctamente:

- SERVIDOR:

  Debemos descargar e instalar el servidor de SmartFox en la máquina que queramos utilizar como servidor de aplicaciones. Podemos acceder a la página de descargas de SmartFoxServer (http://www.smartfoxserver.com/download/sfsPro#p=installer) para obtener el instalador que se corresponda con nuestro sistema operativo y seguir las instrucciones de instalación.

  En la carpeta raíz donde se encuentra el servidor de SmartFox (/SFS/Server) debe haber un archivo de configuración xml (config.xml) donde se definen las distintas zonas de movimiento (habitaciones) que va a tener el usuario dentro de un juego. También se van a definir en ese archivo, la extensión de la base de datos para poder realizar las operaciones sobre la base de datos, y el manejador de la base de datos, que incluye datos como el número máximo de conexiones permitidas, el login y usuario de la base de datos, y el driver utilizado, así como la dirección donde se encuentra la base de datos.

```
- <Zone name="osExample1" uCountUpdate="true" maxUsers="200" customLogin="false">
  - <Rooms>
      <Room name="Limbo" maxUsers="1000" limbo="true" autoJoin="true" isTemp="false" />
    - <Room name="House1Inside" maxUsers="50" isPrivate="false" isTemp="false" uCountUpdate="true">
      - <Vars>
          <Var name="_os_mapId" type="s" private="false">osExample1#m1_House1Inside</Var>
        </Vars>
      </Room>
    - <Room name="House1Outside" maxUsers="50" isPrivate="false" isTemp="false" uCountUpdate="true">
      - <Vars>
          <Var name="_os_mapId" type="s" private="false">osExample1#m3_House1Outside</Var>
        </Vars>
      </Room>
    - <Room name="House2Inside" maxUsers="50" isPrivate="false" isTemp="false" uCountUpdate="true">
      - <Vars>
          <Var name="_os_mapId" type="s" private="false">osExample1#m2_House2Inside</Var>
        </Vars>
      </Room>
    - <Room name="House2Outside" maxUsers="50" isPrivate="false" isTemp="false" uCountUpdate="true">
      - <Vars>
          <Var name="_os_mapId" type="s" private="false">osExample1#m4_House2Outside</Var>
        </Vars>
      </Room>
    </Rooms>
  - <Extensions>
      <extension name="__$OpenSpace$__" className="com.smartfoxserver.openspace.OpenSpaceExtension" type="java" />
      <extension name="dbAvatar" className="dbAvatarExtension.as" type="script" />
    </Extensions>
  - <DatabaseManager active="true">
      <Driver>org.gjt.mm.mysql.Driver</Driver>
      <ConnectionString>jdbc:mysql://localhost:3306/avatar</ConnectionString>
      <UserName>root</UserName>
      <Password>Javiola</Password>
    - <TestSQL>
        <![CDATA[ SELECT idchild from childs where idchild=1  ]]>
      </TestSQL>
      <MaxActive>10</MaxActive>
      <MaxIdle>10</MaxIdle>
      <OnExhaustedPool>fail</OnExhaustedPool>
      <BlockTime>5000</BlockTime>
    </DatabaseManager>
    <AutoReloadExtensions>true</AutoReloadExtensions>
  </Zone>
```

Ilustración 81. Configuración servidor

Tal y como hemos configurado la extensión de la base de datos, en la carpeta *SFS/Server/sfsextensions* debemos tener la clase que corresponde a dicha extensión y que va a contener los métodos que acceden a la base de datos (en nuestro caso la clase dbAvatarExtension.as).

También se debe configurar la extensión del servidor de Openspace (https://cgs-openspace-example.googlecode.com/svn-history/r4/trunk/OpenSpace2/Server/OpenSpace_Extension_PRO.zip) para que el avatar se dibuje correctamente según se ha definido. La extensión de Openspace, viene configurada en la carpeta *SFS/Server/Openspace* en el fichero OpenSpace_server.xml. En dicho fichero, debemos asignar la clase que extiende de avatar y que ha sido creada en la secuencia de movimiento

del avatar como antes se ha explicado para poder gestionar las posiciones y movimientos del avatar. (En nuestro caso llamada avatarDrawer)

```
- <AvatarTypes use8Directions="1">
    <AvatarType mainClass="avatarDrawer" ghostClass="ExampleGhost" stature="72">avatarDrawer</AvatarType>
  </AvatarTypes>
```

Ilustración 82. Configuración extensión

Donde use8Directions indica si el avatar se puede mover en 8 direcciones (nuestro caso) o solo se puede mover en 4 direcciones, **mainClass** es la clase que contiene los métodos de obtención del avatar, **ghostClass** es la clase que contiene los métodos de obtención de la señalización del avatar y **stature** es el tamaño del avatar que se tendrá en cuenta a la hora de poder pasar por túneles y espacios que tengan una altura reducida

- APLICACIÓN:

La aplicación que se quiera ejecutar, debe estar situada en la dirección destinada a las aplicaciones del servidor de SmartFox (SFS\Server\webserver\webapps\root) y para proceder a su ejecución debe ser llamada indicando la ip y el puerto de la siguiente manera (http://127.0.0.1:8080/Avatar/ en nuestro caso).

- BBDD

Para la base de datos, nos descargaremos e instalaremos MySql (http://dev.mysql.com/downloads/file.php?id=455548) y seguiremos las instrucciones de instalación.

Una vez instalada la base de datos, debemos ejecutar el siguiente script, para crear las tablas necesarias para el funcionamiento de la aplicación.

create database avatar;

connect avatar;

create table games (idgame INTEGER NOT NULL , game TEXT NOT NULL, PRIMARY KEY (idgame));

create table instance (idinstance BIGINT NOT NULL, idgame INTEGER, date DATETIME, PRIMARY KEY (idinstance), FOREIGN KEY (idgame) REFERENCES games(idgame));

create table childs (idchild BIGINT NOT NULL, child VARCHAR(50), pass TEXT, avatar TEXT, PRIMARY KEY (idchild));

create table scores (idscore BIGINT NOT NULL, idinstance BIGINT, idchild BIGINT, score NUMERIC, PRIMARY KEY (idscore), FOREIGN KEY (idinstance) REFERENCES instance(idinstance), FOREIGN KEY (idchild) REFERENCES childs(idchild));

7. Herramientas y lenguajes utilizados

Pequeña descripción de las herramientas utilizadas para la realización del proyecto.

7.1 Adobe Flash

Creado por Macromedia, aunque distribuida ahora por parte de Adobe, Flash es una herramienta orientada en sus principios a mostrar animaciones en 2 dimensiones e incluirlas en páginas web, pero que ha ido evolucionando hasta llegar a convertirse en una herramienta de desarrollo completa que permite no solo mostrar animaciones y películas, sino que permite que estas sean interactivas.

Actualmente se ha convertido en una de las herramientas preferidas para desarrollos cuyo destino es la inclusión en una página web, gracias a que la utilización de gráficos vectoriales permite disminuir el tiempo que necesita la aplicación para cargarse en la página web correspondiente.

Otra característica importante de Flash, es que permite trabajar en capas distintas con los objetos que posteriormente van a componer el clip/aplicación, lo que ayuda a la encapsulación de tareas.

La facilidad que aporta el uso de las capas así como el hecho de que la utilización de gráficos vectoriales disminuya el tiempo requerido para la carga de la aplicación, son los principales motivos por los que se ha decidido la utilización de Flash Professional como herramienta de programación.

Para este proyecto en concreto, se ha utilizado la versión Adobe Flash Professional CS5.

Ilustración 83. Versión de Flash Professional utilizada

7.2 ActionScript

ActionScript es el lenguaje de programación utilizado por Flash Professional (en este caso, en su última versión, 3.0). Basado en el estándar ECMAScript [20], ha pasado de ser en su creación un lenguaje muy básico a convertirse en un lenguaje avanzado con soporte de programación orientada a objetos. (ActionScript 3.0)

Desde sus inicios, este lenguaje nació con el objetivo de crear animaciones con audio y video, lo que unido a la compatibilidad de ActionScript 3.0 con la programación orientada a objetos, permite cambiar fácilmente el comportamiento de la aplicación modificando simplemente una o varias clases de ActionScript.

Esa compatibilidad con la programación orientada a objetos, permitirá, en este proyecto, facilitar la comprensión, el mantenimiento y la posible ampliación de código por parte del desarrollador.

7.3 Adobe Photoshop

Creada por Adobe, se trata de una de las herramientas más utilizadas para el tratamiento de imágenes, tanto a nivel profesional como a nivel doméstico. Es conocido popularmente por el retoque fotográfico (portadas, revistas, etc...), aunque sus

posibilidades no se limitan solo a ese campo sino que van más allá, pudiendo generarse gráficos, crear imágenes desde cero o incluso manipulación de clips de video.

En este caso se ha decidido su utilización por la posibilidad de trabajar con diferentes capas a la hora de crear las imágenes originales de los avatares, ya que esto nos permitía crear una base de dibujo e ir creando sobre esa base los nuevos dibujos sin necesidad de hacer todos los dibujos desde cero.

Aprovechando las utilidades de Photoshop, las imágenes se han creado como mapa de bits, para su mejor utilización posterior en la integración de los dibujos tanto dentro de la aplicación, como dentro del motor de juegos.

La capacidad de Photoshop de poder modificar los dibujos a nivel de pixel, ha ayudado a que los movimientos de los avatares en el motor de juegos no sean tan bruscos y sean más definidos de lo que podrían haber sido en el caso de no haber podido trabajar con esa característica.

Para este proyecto en concreto se ha utilizado una versión portable de Photoshop CS4 Extended.

Ilustración 84. Versión de Photoshop utilizada

7.4 OpenSpace

OpenSpace es un marco de trabajo para Flash para el desarrollo de aplicaciones multiusuario y MMO. En conjunto con ActionScript y SmartFoxServer, permite al desarrollado crear un mundo virtual propio en el que poder interactuar.

Ofrece, a través de una perspectiva isométrica, obtener con imágenes 2D la sensación de 3D, como se puede comprobar en multitud de juegos que ya utilizan este motor como "The Mapoosa Clan"[16] o "WinxClub:The Online Game"[17].

[16] http://www.mapoosa.com
[17] http://www.winxclub.com/es

Ilustración 85. Imagen de "The Mapoosa Clan"

Permite la creación y modificación de mapas (incluso en tiempo de ejecución), así como el ángulo de cámara que se desea obtener. A la hora de creación de mapas, utiliza un sistema de capas, por lo que no es necesario crear el mapa de una sola vez, sino que se pueden crear fondos e ir superponiendo uno encima de otro.

Da la posibilidad de visualizar, no solo la parte del panel donde se encuentra el usuario, sino que permite desplazarse a través del mapa mediante cursores y hacer zoom si se desea.

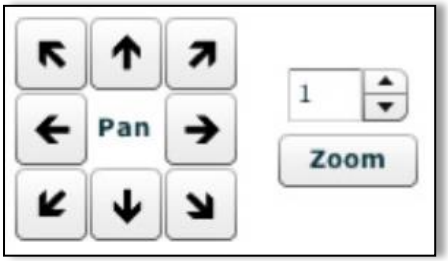


Ilustración 86. Desplazamiento y zoom

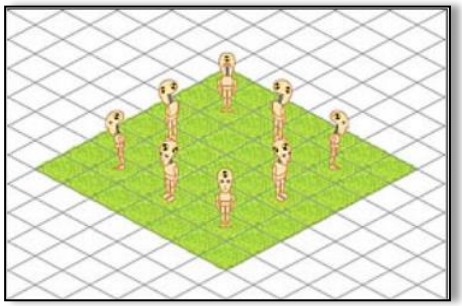


Ilustración 87. Posiciones del Avatar

Hace transparente al desarrollador el algoritmo de desplazamiento del avatar (del que se encarga el motor) y permite la inclusión de cualquier tipo de avatar siempre y cuando esté definido en un clip de película (para lo que utilizamos Flash).

Contiene una extensión dedicada dentro de SmartFoxServer, que se puede utilizar para personalizar las acciones del motor según desee el desarrollador para la funcionalidad del juego.

7.5 SmartFoxServer

Creada por la compañía italiana gotoAndPlay, SmartFoxServer es una plataforma destinada a ayudar en la creación y desarrollo de aplicaciones multiusuario, en concreto MMO.

SmartFoxServer proporciona al desarrollador un conjunto de herramientas que permiten crear aplicaciones sin necesidad de programación alguna en la parte del servidor. Está diseñado en un sistema por capas, que permiten en el caso de que así se necesite, actuar sobre una capa en concreto para usar/modificar las herramientas que dicha capa nos proporciona.

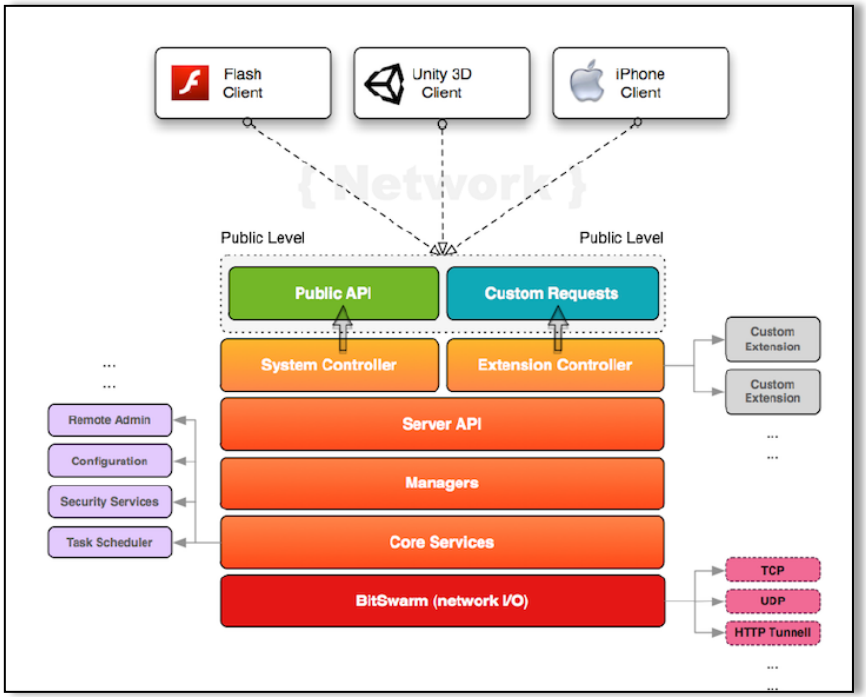
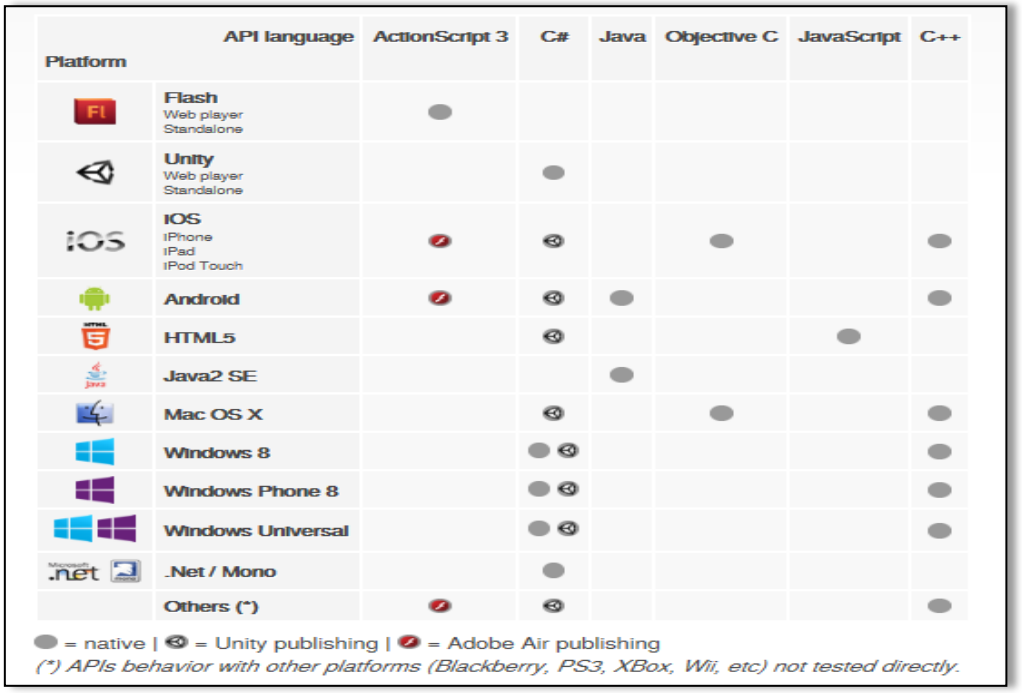


Ilustración 88. Sistema de capas de SmarFoxServer

Además, soporta diferentes plataformas y lenguajes, de modo que varias aplicaciones creadas en distintos lenguajes y entornos pueden interactuar entre sí a través de SmarFoxServer.



Platform	API language	ActionScript 3	C#	Java	Objective C	JavaScript	C++
	Flash Web player Standalone	●					
	Unity Web player Standalone		●				
iOS	iOS IPhone IPad IPod Touch	⊘	◈		●		●
	Android	⊘	◈	●			●
	HTML5		◈			●	
	Java2 SE			●			
	Mac OS X		◈		●		●
	Windows 8		● ◈				●
	Windows Phone 8		● ◈				●
	Windows Universal		● ◈				●
.net	.Net / Mono		●				
	Others (*)	⊘	◈				●

● = native | ◈ = Unity publishing | ⊘ = Adobe Air publishing

(*) APIs behavior with other platforms (Blackberry, PS3, XBox, Wii, etc) not tested directly.

Ilustración 89. Plataformas y lenguajes soportados

SmartFoxServer organiza a los usuarios en zonas, que corresponderían a los distintos juegos que pueden encontrase en el servidor, y dentro de cada zona, pueda haber una o varias salas o habitaciones, que son las entidades más importantes dentro de la arquitectura de SmartFoxServer, ya que cada sala o habitación, representa una aplicación individual en sí misma, y varias salas de una misma zona, pueden estar interconectadas.

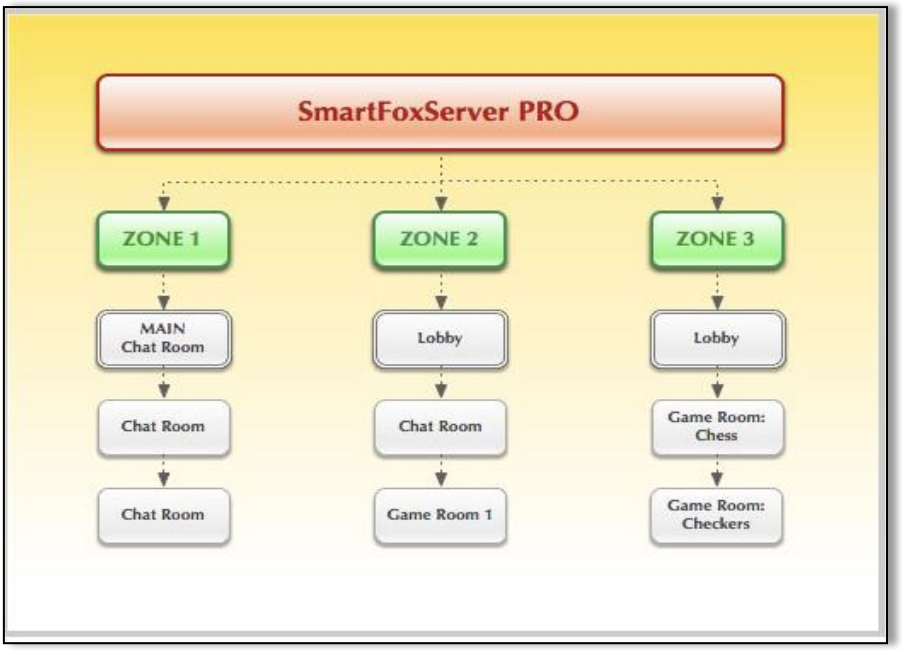


Ilustración 90. Arquitectura de la aplicación

7.6 MySql

Desarrollado por MySQL AB, MySQL es un sistema de gestión de base de datos, relacional, multihilo y multiusuario. Se trata del sistema de gestión de base datos más utilizado actualmente con una distribución total de más de 100 millones [21].

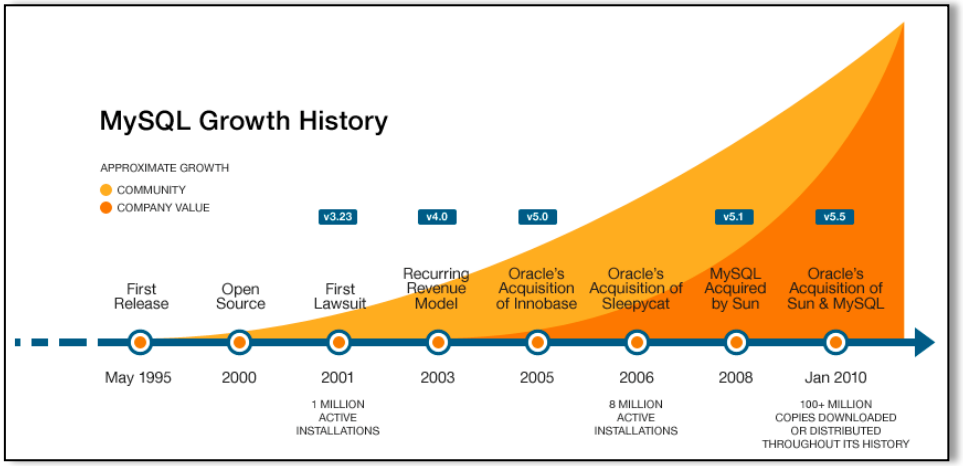


Ilustración 91. Crecimiento de MySQL

Su condición de OpenSource (recordemos que al ser OpenSource, su descarga es gratuita) ha facilitado el hecho de que se trate popularmente del sistema más utilizado, pero también ofrece la posibilidad de la adquisición de licencias para su uso en aplicaciones que no sean OpenSource.

Pero su coste cero no es el único motivo por el que se ha convertido en el sistema de gestión de base de datos más utilizado. Las principales características que definen MySQL son las siguientes:

- es independiente de la plataforma utilizada.
- soporta múltiples usuarios conectados al mismo tiempo (multiusuario).
- puede trabajar simultáneamente en varios procesadores (multihilo).
- es considerada una base de datos muy rápida.
- presenta una gran facilidad de uso.
- soporta la conexión con todos los lenguajes que puedan utilizar la interfaz ODBC, teniendo, para los principales lenguajes de programación, interfaces ya predefinidas.
- está completamente preparado para el trabajo en red, las bases de datos pueden ser accedidas desde cualquier lugar.

Para este proyecto, se ha utilizado la versión disponible de MySQL Workbench 6.2, que contiene MySQL Server 5.6.

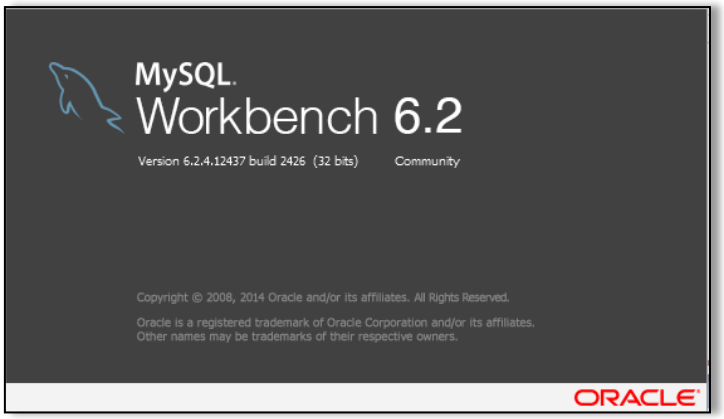


Ilustración 92. Versión de MySQL Workbench

8. Gestión del proyecto

Para la realización del proyecto se ha elegido XP (Extreme Programming) como metodología. Al tratarse de un proyecto realizado únicamente por una persona, las tareas no se solapan nunca, ya que no se dispone de recursos para ello.

A continuación se muestra una lista de las tareas realizadas. La fase de implementación/pruebas se encuentra integrada por utilizar la metodología XP, que supone la prueba constante sobre cada iteración.

Las tareas se agrupan en cuatro fases:

- **ANÁLISIS:** En esta fase se incluye desde las primeras reuniones con el tutor para conocer la idea del proyecto, hasta la investigación de las herramientas con las que había la posibilidad de desarrollar el proyecto, así como la toma de requisitos y la especificación de los mismos.
- **DISEÑO:** Fase que recoge toda la información de la fase de análisis para poder tener claro la estructura de la aplicación, de la base de datos y el estilo de avatar que se desea.
- **IMPLEMENTACIÓN/PRUEBAS:** Al utilizar como metodología XP, en esta fase se realizan simultáneamente la implementación y las pruebas.
- **DOCUMENTACIÓN:** Esta fase comprende la realización del presente documento.

TAREA		DURACION
ANÁLISIS		**312 h**
	Recogida de información	64 h
	Investigación herramientas de desarrollo	160 h
	Identificación de requisitos	48 h
	Especificación de US (User Stories) y casos de uso	40 h
DISEÑO		**96 h**
	Diseño de la BBDD	24 h
	Diseño de la estructura de la aplicación	40 h
	Diseño del avatar	32 h
IMPLEMENTACIÓN/PRUEBAS		**472 h**
	Prototipo del Avatar	16 h
	Navegación de pantallas	16 h
	Creación de la BBDD	8 h
	Crear Usuario	24 h
	Acceso a la aplicación	16 h
	Almacenar Avatar	24 h
	Consulta Avatar	16 h
	Dibujo de características de Avatar para aplicación	80 h
	Dibujo de características en movimiento.	200 h
	Creación de clips de movimiento.	24 h
	Integración de Avatar en motor de juegos.	48 h
DOCUMENTACIÓN		**320 h**
	Generación de la memoria.	320 h

Tabla 1. Estimación del proyecto

A continuación podemos ver la sucesión temporal de tareas en un diagrama de Gantt.

.

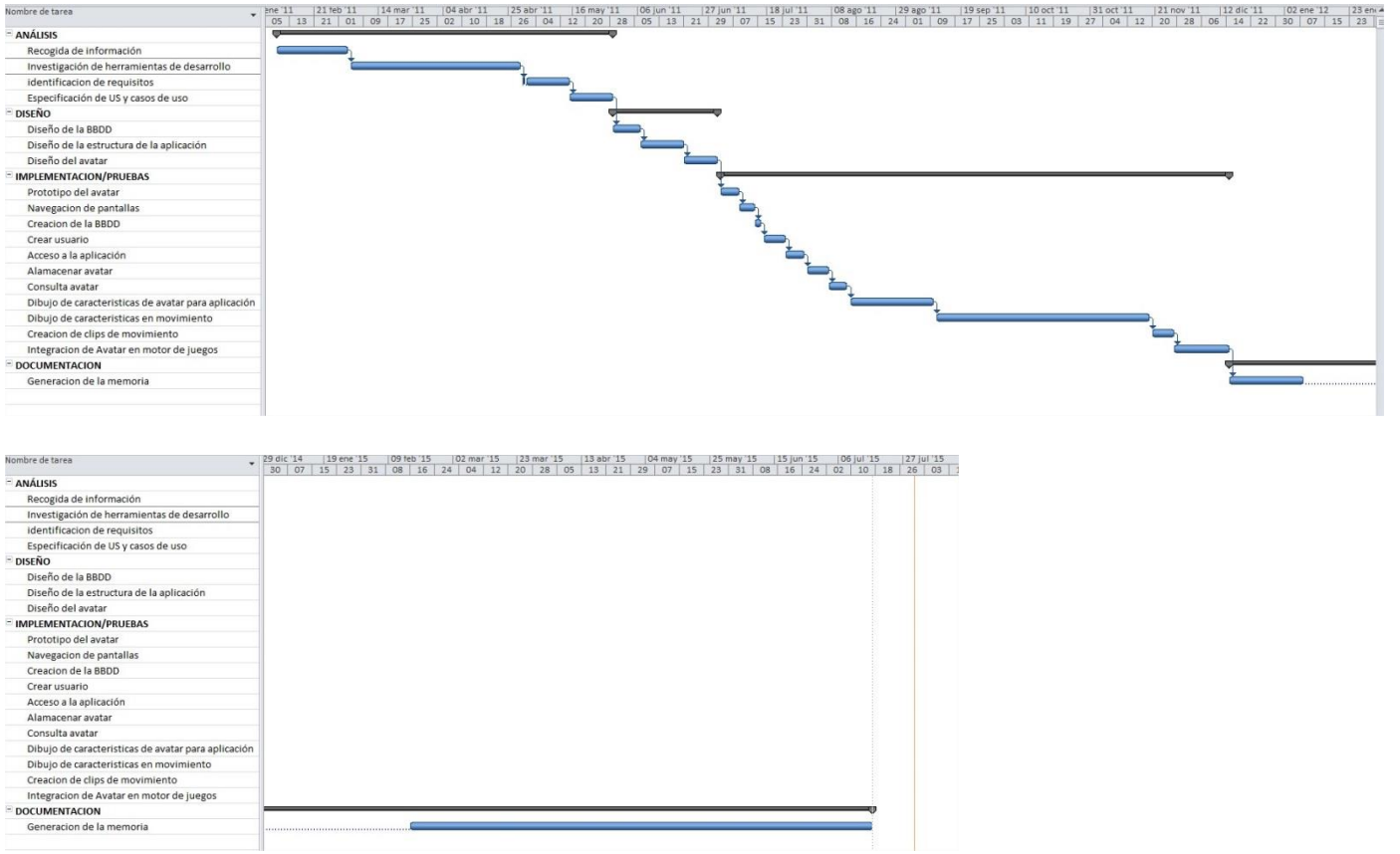


Ilustración 93. Diagrama de Gantt del proyecto

8.1 Presupuesto

Para la evaluación del coste del proyecto se ha tenido en cuenta la utilización de los siguientes recursos humanos:

- Analista. Se encargará de las fases de análisis, diseño y documentación.(42€/h)
- Programador. Se encargará de la fase de implementación/pruebas.(35€/h)

También hay que contar con la utilización de las herramientas de desarrollo, así como el material utilizado.

- Licencias de aplicaciones: Adobe Flash, Adobe Photoshop.
- Herramienta de trabajo: Ordenador portátil, conexión a internet.

El desglose del presupuesto sería el siguiente:

RECURSO	COSTE
Analista (728 h)	30576.00€
Programador (472 h)	16520.00€
Toshiba Satellite Pro C50-A-1L6	617.87€
Conexión internet 5 meses (Fibra ONO 50MB)	188.00€
Adobe Flash (3 meses)	72.57€
Adobe Photoshop (3 meses)	72.57€
TOTAL	**48047.01€**

A este total, habría que incluirle el beneficio esperado, el cual situamos en un 25%, por lo que el resultado total sería el siguiente:

	COSTE
Coste recursos (humanos y materiales)	48047.01€
Beneficio (25%)	12011.75€
TOTAL	**60058.76€**

9. Conclusiones y líneas futuras

Analizando las tareas realizadas durante este proyecto, podemos extraer las siguientes conclusiones que detallamos a continuación, así como una lista de posibles líneas futuras a tener en cuenta en la aplicación.

9.1 Conclusiones

Tras el estudio de las distintas metodologías ágiles y su aplicación (n nuestro caso de XP) en el desarrollo de nuestro proyecto, sea podido observar que todas ellas dinamizan ampliamente los proyectos, aunque también es cierto que actualmente no se encuentran del todo introducidas en el mundo laboral en España. Por lo que se ha podido observar en los distintos artículos leídos, las empresas no optan por una metodología en exclusiva, sino que en su mayoría, cuando optan por las metodologías agiles, combinan una gestión de SCRUM con un desarrollo de XP

En cuanto a los avatares y su importancia en el momento actual, podemos comprobar como cada vez nos sumergimos más y más en redes sociales, mundos virtuales y multitud de aplicaciones que nos hacen interactuar con multitud de gente a la que no conocemos en persona y cuya imagen obtenemos a través de sus distintos perfiles y avatares, al igual que la obtienen ellos de nosotros. Es por ello que actualmente nos encontramos en un momento en el que cada pequeño cambio en nuestra vida, en nuestro estado de ánimo, etc.., nos lleva inmediatamente a reflejarlo de cara al exterior a través de un cambio en nuestros perfiles y avatares.

Estas situaciones, que son habituales en el mundo adulto, son aplicables totalmente al universo infantil, ya que al fin y al cabo, los niños no son más que un reflejo de lo que ven en los adultos, y es ahí, en esa constante necesidad de actualización donde se ve más útil la aplicación realizada en este proyecto.

Por último, en lo que respecta a la inmersión del alumno en los videojuegos educativos, durante la realización de este proyecto y una vez conocido el término “estado de flujo”, se ha podido certificar, que cuando alguien se encuentra dedicado

a una tarea y se introduce en ella hasta el punto de lograr abstraerse por completo del mundo exterior, todo fluye sin aparente esfuerzo y van apareciendo las soluciones como si fuésemos guiados hacia ellas, lo que hace de los juegos educativos, una potente herramienta para el aprendizaje, de manera que sin darse cuenta, el alumno puede almacenar una gran cantidad de conocimientos y hábitos que de intentar proporcionárselos por otros medios más tradicionales de enseñanza, les sería más tedioso y existiría la posibilidad de un abandono o dejadez por parte del alumno.

9.2 Líneas futuras

Una vez realizada la aplicación y después de haber sido testeada y utilizada durante un tiempo, se advierten algunas mejoras de cara a la seguridad, usabilidad y detalle de la misma que podría ser implementadas en un futuro.

Características de personalización: Para aumentar el detalle del avatar, se podrían aumentar tanto las características a modificar, como las opciones dentro de característica, de manera que cada usuario pueda lograr un mayor acercamiento a lo que desea obtener como avatar, como por ejemplo, añadir complementos del tipo de gorros, gafas, tatuajes, etc.…

Número de perspectivas: Para mejorar la definición del avatar en el motor de juegos, se podrían aumentar el número de perspectivas utilizadas, de modo que el avatar pudiese tener la posibilidad de movimiento casi real de 360°.

Múltiples avatares por usuario: Con el fin de poder utilizar el avatar en distintos juegos, o bien para poder mantener diversos avatares almacenados y elegir en función de la decisión del usuario de usar uno u otro, se podría dotar a la aplicación de la capacidad de mantener almacenados un número limitado de avatares por usuario, definiendo uno como principal para ser utilizado y que esta decisión pudiese ser modificada a petición del usuario en cualquier momento desde la aplicación.

Aplicación offline: Para dar la opción de poder usar la aplicación y generar el avatar sin necesidad de estar conectado a ninguna red ni BBDD, se podría dotar a la aplicación de un modo offline que permitiese generar un avatar y almacenarlo en un fichero que luego podría ser cargado en la aplicación una vez conectada a la red y la BBDD.

Sistema de recuperación de contraseñas: Para evitar que el usuario, en caso de que olvide su contraseña pierda el avatar creado y no pueda volver a usar su nombre de usuario anterior, se podría habilitar en la aplicación un sistema de recuperación de contraseña a través del correo electrónico, de modo que a la hora de darse de alta en la aplicación se le soliciten datos al usuario para, por ejemplo, con la introducción del email de registro en la aplicación, se le puedan enviar al mismo, el usuario y la contraseña utilizados.

10. Bibliografía y referencias

[1] Icaza, Heredia, Borch, "Aprendizaje por inmersión orientado a proyectos: método y resultados", disponible en http://www.mty.itesm.mx/rectoria/dda/rieee/pdf-II/s3/72DECICIcazaotrosFinal.pdf (consultado en 2011)

[2] Mihály Csíkszentmihályi, "Flow: The Psychology of Optimal Experience" (Harper Perennial 1991)

[3] Desarrollo Ágil – Mundo Informático, disponible en http://infow.wordpress.com/category/desarrollo-agil/ (consultado en Junio 2011)

[4] Métrica v3 - Portal de Administración Electrónica, disponible en http://administracionelectronica.gob.es/?_nfpb=true&_pageLabel=P60085901274201580632&langPae=es (consultado en Junio 2011)

[5] Manifiesto por el Desarrollo Ágil de Software, disponible en http://agilemanifesto.org/iso/es (consultado en Junio 2011)

[6] Ingeniería de Software, disponible en http://ingenieriadesoftware.mex.tl/52827_DSDM.html (consultado en Junio 2015)

[7] Metodologías ágiles: Programación Extrema (XP), disponible en http://lordpakus.blogspot.com.es/2013/08/metodologias-agiles-programacion.html

[8] Avatar Face Maker Manga, disponible en http://www.manga.com/content/avatar-face-maker

[9] Avatar Simulator for Gaia, disponible en http://www.tektek.org/dream/

[10] Create your own superhero, disponible en http://marvel.com/games/play/31/create_your_own_superhero

[11] DoppelMe - Free Dynamic Avatars, disponible en http://www.doppelme.com

[12] Face Your Manga, disponible en http://www.faceyourmanga.com

[13] My Avatar Editor. Create and edit personalized Mii avatar characters, disponible en http://www.myavatareditor.com

[14] Pick a Face, disponible en http://pickaface.net/

[15] Portrait Illustration Maker: Free Cartoon Avatar Generator, disponible en http://illustmaker.abi-station.com/index_en.shtml

[16] Scott Pilgrim Avatar Creator, disponible en http://www.scottpilgrimthemovie.com/avatarCreator/

[17] South Park Avatar Creator, disponible en http://www.southparkstudios.com/avatar/

[18] OpenSpace Overview, disponible en http://www.openspace-engine.com/overview (consultado en Abril 2011)

[19] Adobe Flash Professional CS5.5 http://www.adobe.com/es/products/flash.html

[20] ECMAScript http://es.wikipedia.org/wiki/ECMAScript

[21] Una breve cronología de MySQL http://www.solocodigoweb.com/blog/2014/03/04/una-breve-cronologia-de-mysql/